NPS-OR-96-001

NAVAL POSTGRADUATE SCHOOL
Monterey, California

The Time Strike Munitions Optimization Model

by

Kirk A. Yost

January 1996

Approved for public release; distribution is unlimited.

Prepared for: USAF Office of Aerospace Studies
Kirkland AFB, NM 87117-5776

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CA 93943-5000

This report was prepared for the U.S. Air Force Office of Aerospace Studies, Kirtland AFB, NM. Reproduction of this report is authorized.

This report was prepared by:

KIRK A. YOST
Major, USAF

Reviewed by:

FRANK C. PETHO
Chairman
Department of Operations Research

Released by:

GORDON B. SCHACHER
Dean of Research (Acting)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1996	Technical Report

4. TITLE AND SUBTITLE
THE TIME STRIKE MUNITIONS OPTIMIZATION MODEL

5. FUNDING NUMBERS

6. AUTHOR(S)
Yost, Kirk A.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-OR-96001

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Aerospace Studies (AFMC)
3550 Aberdeen Avenue SW
Kirtland AFB, NM 87117-5776

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release, distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The TIME STRIKE munitions optimization model was introduced in 1995 for use by various US Air Force agencies to develop requirements for conventional munitions, to refine operational plans based on the availability of different mixes of munitions, and to assess the effects of procuring different types and quantities of munitions. TIME STRIKE was developed under the sponsorship of HQ US Air Force, the Air Force Studies and Analyses Agency, and HQ Air Combat Command to consolidate and extend three existing munitions optimization models. The report covers both the formulation of the new large-scale linear programming model and extensions to the existing models that were included in TIME STRIKE.

14. SUBJECT TERMS
Munitions, Optimization, Linear Programming, Campaign Models, USAF

15. NUMBER OF PAGES
67

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

ABSTRACT

The TIME STRIKE munitions optimization model was introduced in 1995 for use by various US Air Force agencies to develop requirements for conventional munitions, to refine operational plans based on the availability of different mixes of munitions, and to assess the effects of procuring different types and quantities of munitions. TIME STRIKE was developed under the sponsorship of HQ US Air Force, the Air Force Studies and Analyses Agency, and HQ Air Combat Command to consolidate and extend three existing munitions optimization models. The report covers both the formulation of the new large-scale linear programming model and extensions to the existing models that were included in TIME STRIKE.

EXECUTIVE SUMMARY

In January 1995, three USAF agencies that had been using different optimization models to analyze requirements for conventional aircraft munitions agreed to consolidate and extend their models. The three models consisted of HEAVY ATTACK, operated by the HQ Air Force Deputy Chief of Staff for Operations, Directorate of Forces (HQ USAF/XOFW); the Theater Attack Model (TAM), operated by the Air Force Studies and Analyses Agency (AFSAA); and the MIXMASTER model, operated by HQ Air Combat Command's Plans and Programs Directorate (HQ ACC/XP-SAS). The agencies felt a consolidation would advance the joint capabilities of this class of models, leverage their investment in common databases and data management tools, and unify and reconcile their various analyses.

The USAF Office of Aerospace Studies (OAS), working under the direction of an inter-agency working group, produced a new model called TIME STRIKE that offers a menu of user-selectable objectives and constraints. TIME STRIKE isn't a single model, but instead is a family of optimizations with a common core. In each instance, TIME STRIKE decides how best to allocate aircraft sorties and weapons to targets in a particular scenario, subject to various budget and availability constraints. However, TIME STRIKE differs from its three ancestors in eight major areas:

- New objective functions that are oriented towards campaign goals have been added;
- Sortie and target kill accounting have been changed;
- Time periods are treated explicitly;
- Battle-damage assessment (BDA) and target regeneration have been revised;
- Weather effects have been reformulated;
- Operationally-oriented limitations such as minimum altitudes for weapons deliveries have been added;
- Budget constraints have been revised and extended; and
- The ability to model simultaneous campaigns in two theaters has been included.

The paper describes parts of HEAVY ATTACK, TAM, and MIXMASTER where appropriate.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I use "we" often in this report to remind the reader that the models I've documented are the result of a collective effort. A particularly notable member of the team was 1Lt Jay DeYonke, the principle author of QUICK STRIKE. Many of the concepts in these models are Jay's work, and he made large contributions to this field in a very short time. Jay has since left the analyst ranks and gone on to a new career as an aviator, and I wish him well.

Other important contributors include: Maj J. Q. Watton from HQ ACC, our resident fighter pilot and expert war planner; and Capt Paul Campbell from OAS, who has inherited the care and feeding of the models. Also deserving thanks are the members of the USAF Munitions Model Working Group: LtCol Bob Sheldon, LtCol Paul Schroeder, and Maj Ray Hill from AFSAA; Maj Bob O'Neill from JCS J-8; LtCol Doug Lincoln and Maj Mike Buck from HQ USAF/XOFW; Mr. Rich Freet from HQ ACC/XP-SAS; Ms. Lynne Willis from ASC/XREW; and Mr. Nick Reybrock and Mr. Dennis Coulter from ASI. Top cover was ably provided by the OAS management team of Mr. Jerry Colyer, LtCol Roy Rice, and Mr. Jim Haile. None of this work would have happened without the support of Col Tom Allen from AFSAA, Col Craig Ghelber from HQ ACC/XP-SAS, and Col Frank Griffin from HQ USAF/XOFW; they agreed to set their organizations' models aside and pursue the consolidation that led to QUICK STRIKE and TIME STRIKE.

The accuracy and readability of this report is largely due to Dr. Jerry Brown, Dr. Al Washburn, and Dr. Rob Dell here at the Naval Postgraduate School. Dr. Brown and Dr. Washburn have supported USAF munitions analyses for over 20 years, and they spent a tremendous amount of time reviewing drafts, correcting errors, and suggesting improvements. In particular, their review led to a complete revision of the target regeneration and BDA submodel. Dr. Dell provided an independent scrub of the paper, found the (numerous) errors the rest of us missed, and was instrumental in making the discussion comprehensible to readers not familiar with munitions analysis.

Finally, I would like to acknowledge the numerous developers of HEAVY ATTACK, TAM, and MIXMASTER for their contributions over the last two decades. QUICK STRIKE and TIME STRIKE are third-generation models built from these existing systems, and we have freely borrowed the many good ideas employed by our predecessors.

I. INTRODUCTION

A. THE MUNITIONS OPTIMIZATION PROBLEM

The general problem the US Air Force faces when procuring and managing conventional aircraft munitions is determining the best mix of weapons to hold in inventory. The desire to determine the best inventory—along with the structure of the problem—led the Air Force to adopt optimization over 25 years ago as a means to determine munitions stocks.

However, the Air Force's experience has shown there is a more specific set of problem definitions, with the following three covering virtually all questions a munitions optimization must answer:

- **The Tradeoff Problem.** What is the effect of having or not having a particular weapon in the inventory?
- **The Allocation Problem.** What is the best way to allocate munitions and aircraft to targets, given a fixed inventory and scenario?
- **The Requirements Problem.** What weapons inventories do we need to meet our warfighting goals for a particular scenario?

Over the years the Air Force has built a series of models to address these problems, all of which require certain fundamental inputs. First, the models need a scenario, which consists of a collection of target types of various quantities and some measure of importance or precedence for their destruction. Second, the models require a set of aircraft, which fly time-varying *sortie rates* (missions per aircraft per day). Third, the models need data describing the effectiveness of each feasible aircraft-weapon combination against each target type. Given this information, these models try to optimize the allocation of aircraft and weapons into sorties against targets in accordance with some objective function.

The Air Force has a common approach to the munitions problem: use optimization to best allocate aircraft and weapons to targets in a particular scenario. However, the objective functions and constraints of the existing models differ significantly. There is no general agreement on what the meaning of "best" is, nor is there much agreement on which constraints are necessary.

Before proceeding, it will be helpful to discuss the dimensions of the entities in this class of models (Figure 1). *Sorties* are valid combinations of an aircraft, weapon, weapons loadout, delivery tactic (or profile), time period, weather state, target, and target depth (or distance band). *Targets* are classified by

type, distribution across distance bands, and target class. *Weapons* are characterized by type, component family (for weapons that share common parts), and qualification requirements (for weapons that can only be employed by a limited proportion of aircraft or aircrews). The existing models use these dimensions in varying degrees.

SORTIE

aircraft type (F-16)
weapon type (GBU-24 laser-guided bomb)
loadout (2 GBU-24's)
delivery profile (level delivery at 10,000 ft altitude)
time period (day 1-3)
weather state (12,000 ft ceiling, 5 NM visibility)
target type (aircraft shelter)
distance band (100-150 NM from border)

TARGET

type (aircraft shelter)
distance distribution (40% at 100-150 NM)
target class (airfield targets)

WEAPON

type (GBU-24)
component family (GBU's w/ common laser seeker)
qualification family (aircraft with laser designators)

Figure 1. Dimensions of the primary entities in existing munitions models. The models all use sorties, weapons, and targets, but with different levels of detail. This figure gives an example of each dimension.

B. MODEL CONSOLIDATION

The differences among the existing models led to serious disagreements over weapons requirements, which became harder to reconcile as the Air Force's procurement budgets began shrinking in 1990. In January 1995, three USAF agencies that owned existing models agreed to consolidate their optimizations into one system. The three models consisted of HEAVY ATTACK, operated by the HQ Air Force Deputy Chief of Staff for Operations, Directorate of Forces (HQ USAF/XOFW); the Theater Attack Model (TAM), operated by the Air Force Studies and Analyses Agency (AFSAA); and MIXMASTER, operated by HQ Air Combat Command's Plans and Programs Directorate (ACC/XP-SAS). The agencies felt a consolidation would advance the capabilities of this class of models, leverage their investment in common databases and data management tools, and provide a common framework for their analyses.

A working group gave the USAF Office of Aerospace Studies (OAS) the task of combining and extending the three existing models. Consequently, OAS produced two variants of the same formulations. The first set of models, collectively called QUICK STRIKE, operate as a sequence of optimizations. QUICK STRIKE optimizes sortie allocations for a single period, and passes the output from that period to the next period's optimization. This time-myopic approach keeps the model small and fast, but complicates global analyses and forces the user, rather than the model, to explicitly define how resources can be used across time. The other variants, collectively called TIME STRIKE, globally optimize allocations across

time. This paper documents TIME STRIKE, which is a superset; QUICK STRIKE is identical except for its single-period solution horizon. For details on QUICK STRIKE, see DeYonke (1995).

C. THE EXISTING MODELS: HEAVY ATTACK, TAM, AND MIXMASTER

At this point, it is useful to provide a brief overview of the three existing models included in the consolidation, but we will not describe them in detail. For more information, see Brown, Washburn, and Coulter [1994], Jackson [1989], Might [1987], or Yost [1995].

HEAVY ATTACK is the oldest of the models, having been in use since 1973. The model was originally formulated by analysts in the Office of the Secretary of Defense and was implemented by RAND (Clausen, Graves, and Lu [1974]). HEAVY ATTACK assigns values to each target and optimizes the total target value destroyed (TVD). The model uses a nonlinear objective function to capture battle-damage assessment (BDA) effects, and optimizes for a single period (the time-myopic approach). HEAVY ATTACK is the most aggregated of the three models, allocating aircraft sorties to targets without directly modeling weapons. Instead, HEAVY ATTACK determines the best weapon for each combination of aircraft, target, and weather state and computes a composite effectiveness for an aircraft sortie against a target using an input weather distribution. HEAVY ATTACK also does not model aircraft attrition; available sorties are an input, and the model's allocation does not affect available sorties. HEAVY ATTACK does not contain budget constraints, and only has the single objective of maximizing TVD. The amount of aggregation in the model, along with the use of advanced nonlinear programming techniques, makes HEAVY ATTACK very small and very fast, with response times in seconds.

TAM was developed by the Air Force Studies and Analyses Agency in the mid-1980's. TAM is highly detailed, allocating sorties by aircraft, weapon, target type, target distance, weather state, and time period. In addition, TAM offers multiple objective functions, budget constraints and attrition constraints. The most common TAM objective is maximizing TVD; as opposed to HEAVY ATTACK, all TAM's objective functions are linear. BDA is not modeled in TAM, but available sorties are affected by attrition. TAM weather differs from HEAVY ATTACK in that TAM assumes the weather is known perfectly. However, the model uses the weather distribution to constrain the proportion of the time each sortie type can be used. TAM optimizes globally across time, but this feature and other dimensions in the model make the resulting optimizations very large. TAM solution times range from one to three hours.

MIXMASTER is a collective name for an optimization model and an heuristic developed at the Air Force's HQ Air Combat Command in 1990. The MIXMASTER linear program (LP) is a time-myopic version of TAM with only the TVD objective function, while the MIXMASTER heuristic is a greedy sortie allocation scheme that uses target values to determine the proportion of sorties dedicated to each target type. MIXMASTER was built as a response to dissatisfaction with HEAVY ATTACK, and the developers

were directed *not* to use optimization. The LP version of MIXMASTER was written only as a check for the heuristic (Langbehn and Lindsey [1991]).

Figure 2 summarizes the characteristics of the existing models, and illustrates the wide disparity in model philosophies with respect to objectives, constraints, and dimensionality.

	HEAVY ATTACK	TAM	MIXMASTER
Objective function			
linear		X	X
nonlinear	X		
multiple		X	
Sortie dimensions			
aircraft	X	X	X
weapon		X	X
target	X	X	X
loadout		X	X
time period		X	
distance band		X	X
weather state		X	X
Target dimensions			
type	X	X	X
distance band		X	X
Time approach			
myopic	X		X
global		X	
Miscellaneous			
BDA	X		
weather known		X	X
weather unknown	X		
budget		X	
attrition affects sorties		X	X

Figure 2. Capabilities of existing munitions models. The existing models vary widely with respect to objectives, constraints, and dimensionality.

II. TIME STRIKE OVERVIEW

While TIME STRIKE is intended to be a consolidation, several features have been added that were not available in the existing models. Also, TIME STRIKE is not a single formulation; it offers user-selectable objective functions and constraints to allow the analyst to tailor the model to the issue at hand.

A. OBJECTIVE FUNCTIONS

TIME STRIKE retains some of the objectives common to the three existing models, but modifies them. In addition, TIME STRIKE adds two campaign-oriented objectives, which have proven to be the most popular during development tests at HQ ACC.

The most commonly-used objective in the existing models is maximizing TVD. TIME STRIKE offers this objective, but we do not recommend it to new users because relying on target values to control the campaign is a common criticism of the existing models. While target values make sense from an economic point of view, they have proven difficult to determine in practice and are widely viewed as tuning knobs used to get the existing models to kill targets in a particular order. As a result, the user community tasked the working group to find a more natural way of modeling campaign goals without constantly adjusting target values.

The next two objectives are inherited from TAM. The first minimizes aircraft attrition subject to a set of target destruction goals, while the second objective minimizes the cost of buying new aircraft and weapons subject to target destruction goals. The problem with these objectives is that they are *inelastic*; that is, if the model can't kill all the targets required, the model terminates as infeasible and yields little useful information.

To overcome this problem, TIME STRIKE offers two elastic objectives. The first, called the *time-scripted* objective, allows the user to designate goals for destroying targets across time. TIME STRIKE minimizes the sum of the penalties associated with not achieving the goals, which keeps the model feasibly if the goals can't be met.

The time-scripted objective works well in cases where the user is evaluating a specified schedule for a campaign. However, users often want to determine the time necessary to achieve campaign objectives, so TIME STRIKE's final objective function is called the *phase-goal* objective. In this objective, the user divides the campaign into *phases*, which are sets of goals for each target class. The objective pursues the phases in a hierarchical order defined by the user, and attempts to minimize the time required to accomplish the phases. This objective allows the user to define overlap between the phases, so a phase

can start before all the goals in the previous phase are met. As a result, the user can control each goal's degree of preemption.

TIME STRIKE's notion of target classes is a major difference from the existing models, and supports the fact that campaign objectives involve killing collections of related targets rather than individual target types. Target values cause problems in the existing models because they do not apply to sets of targets; instead, the user has to determine values that induce the model to kill the targets in sets. TIME STRIKE avoids these problems by allowing a user to group a set of target types, set a time- or phase-dependent goal for their destruction, and rely on the model to treat them as a group. An example is shown in Figure 3:

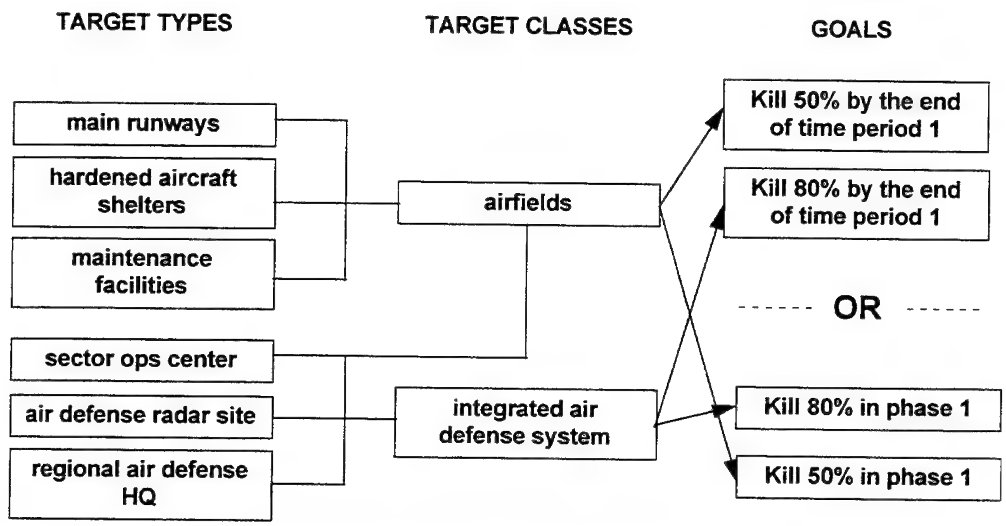

Figure 3. Targets, target classes, and goals in TIME STRIKE. A target can be included in multiple classes, and each class can have its own time- or- phase-dependent goal. User-defined penalties determine the importance of achieving each goal.

In this example, the sector ops center is a member of both the airfield and integrated air defense system target classes. TIME STRIKE's phase goal objective function would require the user to define the proportion of targets in each class that need to be killed to complete the phase, while the other objectives would require the user to set proportions by time period. In any case, grouping targets by class and setting the objectives by class is a much more natural way to express campaign objectives to the model than using individual target values.

B. SORTIE AND KILL ACCOUNTING

TIME STRIKE unifies several ideas in the existing models about what can happen on each sortie and how kills are counted. Figure 4 shows all possible sortie outcomes in TIME STRIKE.

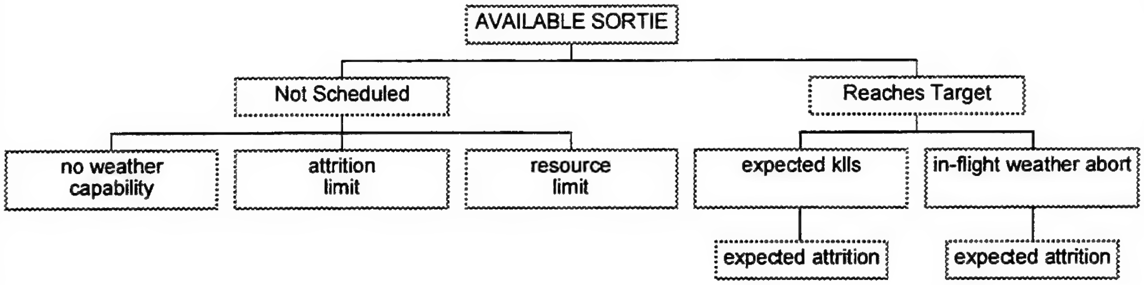

Figure 4. Possible outcomes of a sortie in TIME STRIKE. The available sortie may not be scheduled due to a lack of weather capability, the model already having lost too many aircraft, or the lack of a resource such as weapons. Otherwise, the aircraft reaches the target and is subject to attrition. The model computes expected kills for the cases that do not abort at the target due to weather.

The outcomes are straightforward. A sortie may not be scheduled due to an unfavorable weather forecast, an attrition limit which prohibits further flying, or an aircraft running out of a resource such as weapons. Also, the sortie is subject to a probability of an in-flight weather abort due to errors in the forecast. Expected kills and expected attrition for sorties that strike the target are inputs, and we assume the expected kills are adjusted for the number of aircraft that are killed prior to reaching the target.

Once a target is struck, there are also several possible outcomes, as shown in Figure 5. This kill-accounting scheme captures an important effect—previously modeled only in HEAVY ATTACK—which is that a target can be killed, but misclassified and restruck. This dilution of sorties due to incorrect battle-damage assessment (BDA) is an important effect and must be represented in a realistic model. Another important effect the ability of the enemy to regenerate (repair) dead targets. In both cases, TIME STRIKE has modified and extended the existing approaches, as discussed later.

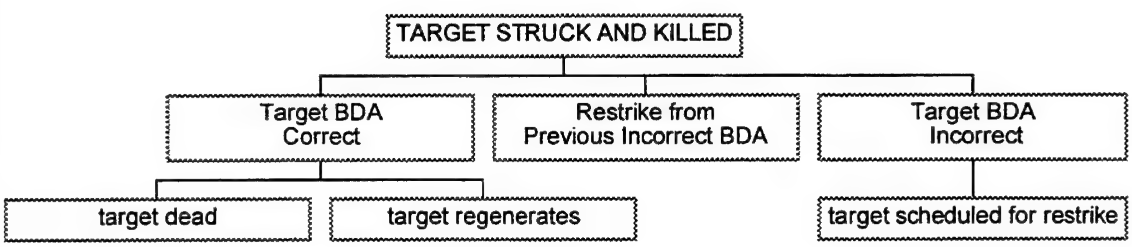

Figure 5. Possible outcomes of a target kill in TIME STRIKE. A kill may either have correct BDA, in which case it is either dead forever or regenerates, or it may have incorrect BDA and is scheduled for a restrike. Only one of the outcomes results in a permanent kill.

C. TIME MODELING

Time periods are necessary to model arrivals of aircraft and weapons in the theater, changes in sortie rates, shifts in campaign objectives, and changes in attrition rates. However, the existing models view time differently. Both HEAVY ATTACK and MIXMASTER are time-myopic, with output from each period's solution (with perhaps some external alteration) used as input for the next period. TAM, on the other hand, has an intrinsic time index in the formulation.

There are disadvantages to adding time to a model. HEAVY ATTACK can use a nonlinear BDA function and still remain small and fast because it only optimizes in a single period. Adding time to HEAVY ATTACK would enormously complicate the model. In addition, there is a good argument for forcing myopia. The existing models conduct one-sided campaigns — the enemy has no choices. Letting an optimization look across time contradicts reality, particularly when the models assume the enemy doesn't react. This omniscience has been a perennial problem in TAM. TAM tends to wait for periods with low attrition rates to kill difficult targets; it also uses its knowledge of the future to kill easy targets with high target values early so more of them are repaired and then restruck (earning more TVD).

Adding time also increases the size of the model. HEAVY ATTACK and MIXMASTER are small and quick, because each period's optimization consists of 1,000-2,000 variables and a few hundred constraints. On the other hand, TAM can grow as large as 180,000 variables and 5,000 constraints due to the intrinsic time index (as well as the weather index, distance index, and so on). This leads to another tradeoff: TAM is usually run with only 4 periods of 3,7, 20, and 30 days, because the LP becomes too big to solve otherwise. Conversely, HEAVY ATTACK can run 20-30 myopic time periods in very little time.

Nonetheless, the myopic approach is a disadvantage for the analyst trying to solve a resource allocation or budgeting problem. If there is a fixed pool of procurement money available for a multi-period scenario, the analyst has to explicitly allocate or constrain expenditures by period. Since optimization is good at making these decisions, it seems unreasonable to force the analyst to guess the best time-constrained allocations outside of the model.

The compromise reached in TIME STRIKE is to use time explicitly in the model, but to limit the optimization's false omniscience. In TIME STRIKE, time is still divided into periods of user-selectable lengths, but now each period consists of an integral number of fixed-length *planning cycles*. A planning cycle is the number of days over which we execute the campaign with no feedback from our actions; in other words, this is the assessment time lag. The planning cycle is key to TIME STRIKE's BDA and target regeneration submodels, because it reintroduces myopia and some of the so-called friction of war into the model. If we could solve enormous models at no cost, we would simply define the time period length as the

planning cycle length. Unfortunately, this isn't possible, so we use the notion of a planning cycle to capture BDA and target regeneration effects within a period.

Figure 6 summarizes the TIME STRIKE's time concepts. The user decides the total length of the campaign and divides the campaign into periods. The periods can be unequal lengths, but each must contain an integral number of planning cycles. If an analyst is using the time-scripted objective, he must specify goals for each target class and time period, and the model will try to meet the goals. If he is using the phase-goal objective, he must determine goals for each target class in each phase, and then the model will try to minimize the number of time periods required to achieve the phases.

The analyst must weigh time fidelity in the model versus responsiveness when using TIME STRIKE. If the analyst needs many time periods for goal changes and aircraft arrivals, he can do so at the cost of generating a much bigger model. If his goals are coarser over time and he needs quicker turnaround, he can use fewer time periods and generate a smaller, faster model. In either case, the addition of the planning cycle cures problems with BDA and target regeneration within a period, as we'll discuss in the next section.

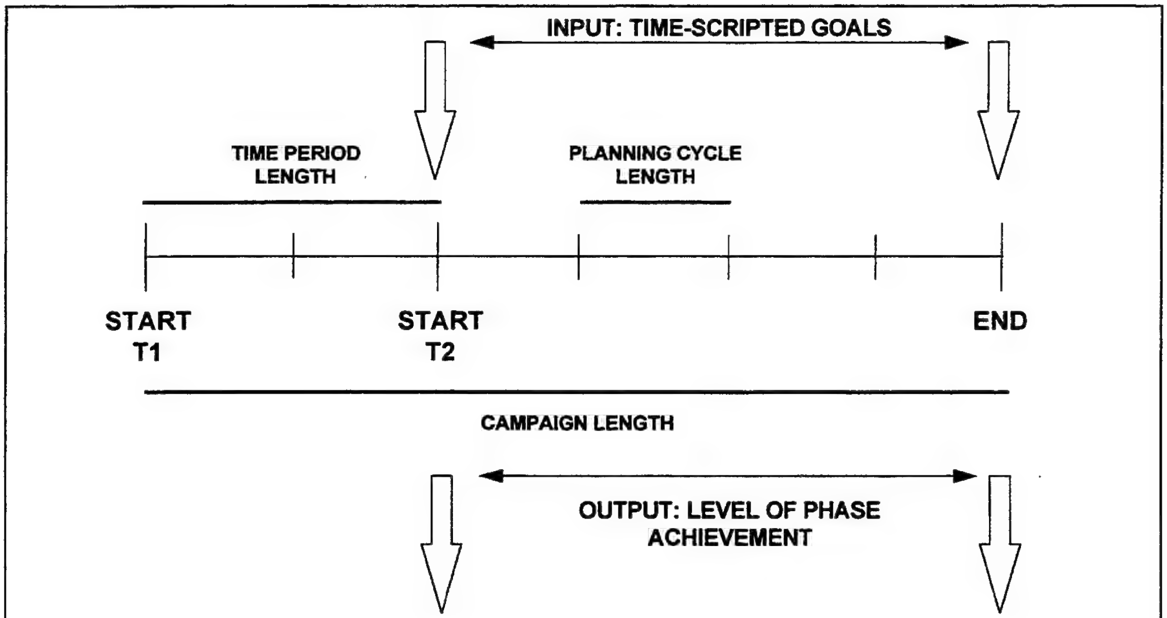

Figure 6. Time periods, planning cycles, and types of goals in TIME STRIKE. Period T1 contains 2 planning cycles, while Period T2 contains 4 planning cycles; the entire campaign consists of these two periods. If the analyst is using the time-scripted objective, he designates goals for the end of each period as input; if he is using the phase-goal objective, the model gives the level of phase achievement by the end of each period as output.

D. BATTLE-DAMAGE ASSESSMENT AND TARGET REGENERATION

BDA has been a problem for the existing models, with HEAVY ATTACK being the only model that accounts for restriking dead targets due to bad BDA. TAM and MIXMASTER assume a dead target is never restruck, which is unrealistic.

In HEAVY ATTACK, the probability of restriking a dead target is a function of the number of targets already killed and a parameter known as the "C-factor," which varies between 0 and 1. A C-factor of 0 implies perfect BDA, while a C-factor of 1 implies no BDA and random targeting. There are two problems with this approach. First, the C-factor has no physical meaning. C-factors are not probabilities, but are merely adjustment factors that determine the marginal returns of continually attacking a particular set of targets (Lord [1982], Boger and Washburn [1985]). As a result, the analyst has to set the C-factors based on their effects on the model output rather than by using any available data.

Second, HEAVY ATTACK presumes that success in killing additional targets of a particular type is a function of the number of those types of targets already killed. For a collection of tanks on a battlefield in a short time interval, HEAVY ATTACK's BDA scheme is a good model. The more tanks that are killed, the more difficult it is for an attacker to discriminate among live and dead tanks. On the other hand, this is not a good model for fixed targets such as bridges. For these targets, the probability of a bad assessment has nothing to do with the number of similar facilities that have been bombed.

QUICK STRIKE originally used a linearized version of a BDA model proposed by Boger and Washburn [1985] that recast the C-factors as the probability of correctly assessing a live target. Boger and Washburn developed a differential equation relating this probability to the probability of killing a live target, given a certain number of live and dead targets. The problem with this approach is it has too much memory. A tank killed on the battlefield 30 days ago probably has little or no effect on the current assessment issues: the battlefield has probably moved, and the dead tank no longer functions as a decoy.

The BDA problem is an open research issue. In the meantime, TIME STRIKE's BDA model is a compromise that keeps the model linear, explicitly defines the BDA factors, and denies the optimization's tendency to defeat BDA effects through omniscience. First, TIME STRIKE uses a single BDA input for each target type, which is a static probability of misclassifying a dead target as still being alive. Second, TIME STRIKE does not allow credit for any more kills against that target type until each misclassified target is restruck. This is illustrated in Figure 7.

In this example, T1 contains 2 planning cycles. We assume kills occur uniformly across a time period, so half of the misclassified targets happen in the first planning cycle of T1 and must be restruck in

T1, while the other half must be restruck in T2. On the other hand, T2 contains 4 planning cycles, so 3/4 of the targets struck in the period that have incorrect BDA must be restruck within T2.

This mechanism allows us to capture the BDA effects *and* the lag effects in long time periods. If the model could wait until the next period to restrike targets, it would tend to wait until the *last* period to accumulate kills and avoid the workload caused by bad BDA. This can't happen in TIME STRIKE, as kills against these targets are discounted and the model prohibits additional kills against other targets until the bad BDA workload is accomplished. Conversely, the planning cycle lag forces some semblance of reality by making the model wait to recognize the need to do restrikes.

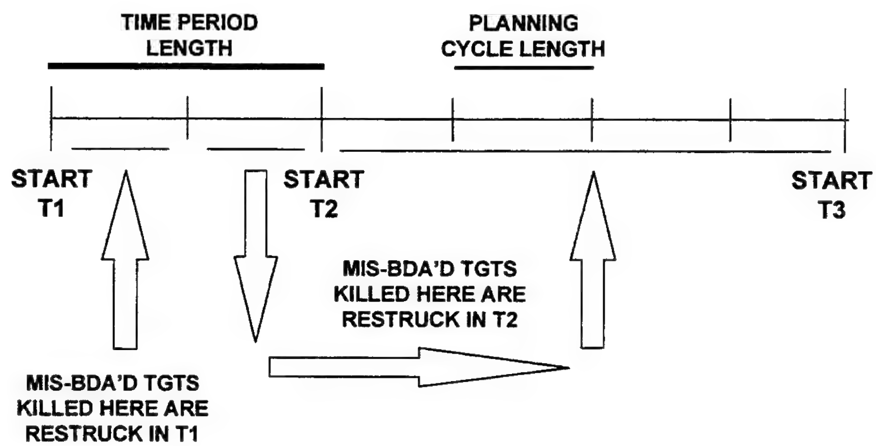

Figure 7. BDA in TIME STRIKE. Period T1 contains 2 planning cycles. Incorrect BDA from targets struck in the first planning cycle of T1 forces restrikes in the second planning cycle of T1, while incorrect BDA in the second planning cycle of T1 forces restrikes in period T2. This approach allows for both a lag time to recognize bad BDA and the ability to schedule restrikes within a time period.

Target regeneration also uses the planning cycle. TIME STRIKE lags the detection of regenerated targets by one planning cycle, using the same logic as it uses for incorrect BDA. Again, the assumption of the planning cycle is that the sortie allocation is fixed over the length of the cycle, and the model cannot act on new information until the next cycle. Therefore, a newly-regenerated target must wait one cycle before it can be retargeted.

A serious limitation of the existing models is that they do not allow target regeneration within a period, which is a problem for targets with short repair times in long time periods. For example, a target such as a runway may have a 12-hour regeneration time. Unfortunately, the existing models will kill it once in a 30-day period, assume it's suppressed the entire time, and not detect it as functional until the next

period. This seriously overestimates the progress the model is making in the scenario, and, in the case of TAM, makes the optimization more likely to kill targets in long time periods. TIME STRIKE allows multiple regenerations and retargeting within a time period, up to the number of planning cycles. Since the objective functions only count the target's status at the end of a time period, TIME STRIKE must allocate more sorties than the existing models to keep a target dead or in repair.

TIME STRIKE can also control the total number of targets regenerated. TAM assumes every target that is killed is eventually repaired, while the version of HEAVY ATTACK currently in use only allows targets to regenerate once. TIME STRIKE compromises by using an input repair proportion to determine the expected number of targets repaired after every planning cycle; in addition, the user can also adjust this parameter to implicitly constrain total repair capacity.

Target regeneration and BDA are closely related in TIME STRIKE, and are implemented in one submodel. The formulation of this submodel is quite lengthy and is presented from a stochastic point of view; the details are in Appendix A.

E. WEATHER EFFECTS

The existing models describe weather in terms of "weather states", which are mutually exhaustive combinations of ceiling and visibility. Historically, the munitions-analysis community has partitioned the distribution of weather into 6 states and has used the proportion of the time the weather is in each state as a static input. These states affect the model because each aircraft-weapon combination has a number of *delivery profiles* associated with it, and each profile is valid only in certain weather states. For example, a medium-altitude profile might only be possible in the best three weather states, while a low-altitude profile using radar bombing might be possible in any weather state.

TAM and MIXMASTER assume perfect weather knowledge. There is no sense of a forecast, and these models assume the weather states occur in their fixed proportions in each period. On the other hand, HEAVY ATTACK models weather through its weapon-aggregation scheme. The HEAVY ATTACK preprocessor first finds the best weapon and delivery profile for every aircraft-target-weather state combination, and then computes a weighted average of weapons effects and attrition for each aircraft-target combination based on the weather distribution. This procedure is equivalent to assuming that the weather is unknown when aircraft are allocated to targets, but known when weapons and delivery profiles are selected.

Neither assumption is true. We don't have perfect weather knowledge, but we can forecast with some degree of accuracy. This issue has become more important as we develop autonomous (and expensive) weapons that have guidance systems unaffected by weather, because we need to correctly

measure the payoff from having such weather-resistant weapons. Unfortunately, weather is a difficult issue, and weapons are affected by more that just ceiling and visibility. Humidity, precipitation, fog, thermal contrast, infared transmittance, and many other factors affect the ability to deliver a weapon. In addition, modeling weather suggests a Markov decision process (e.g., Ross [1993]): first, we make a decision based on a forecast; then, nature acts; and then we revise our decisions based on the outcome of our former decision and the subsequent state of nature.

Weather effects are an open research question for these models. TIME STRIKE does not offer a complete solution to the weather problem, but takes a step further than existing models by forcing sorties to be scheduled in accordance with the average forecast rather than the distribution of weather encountered. More importantly, TIME STRIKE dilutes scheduled sorties by using the probability of forecast error to determine the number of in-flight weather aborts.

As an example, consider the following data provided by the Air Force Environmental Technical Applications Center in Figure 8.

WEATHER STATE (WX)	PROBABILITY OF FORECAST
WX 1	0.020
WX 2	0.050
WX 3	0.040
WX 4	0.031
WX 5	0.053
WX 6	0.806
TOTAL	1.000

Figure 8. Marginal forecast probabilities by weather state. Higher numbers indicate more favorable weather. For example, WX 1 represents a ceiling of 0 feet and 0 NM visibility, while WX 6 represents a ceiling of 12000 feet and a 5 NM visibility.

A delivery profile that is valid in one weather state (WX) is valid in any higher state. Therefore, TIME STRIKE selects delivery profiles based on the forecast using cumulative constraints. In this example, 2% of the profiles must be capable in WX 1, 7% must be capable in WX 1 or WX 2, 11% must be capable in WX 1, WX 2, or WX 3, and so on. If an aircraft type is only capable in WX 6, then TIME STRIKE assumes 19.4% of the available sorties for that aircraft type are lost in the period; these sorties are unscheduled and are not subject to attrition.

To account for forecast error, we use the conditional probability the weather is invalid for the profile, given that we forecast valid weather for the profile. We use this probability to determine the proportion of sorties that aren't aborted in-flight due to weather; data for our example is shown in Figure 9.

WX REQUIRED FOR PROFILE	PROPORTION NOT ABORTED
WX 1 OR BETTER	1.0000
WX 2 OR BETTER	0.9502
WX 3 OR BETTER	0.8596
WX 4 OR BETTER	0.8185
WX 5 OR BETTER	0.7675
WX 6	0.7665

Figure 9. Non-abort proportions by weather state. These are the proportion of the time the weather is in a particular state or better, given the forecast was for a particular state or better.

Suppose the model uses a profile whose minimum weather state is WX 3. Given the forecast was for WX 3 or better, there is a .8596 probability the weather will be WX 3 or better. When the model schedules WX 3 deliveries based on a forecast, 85.96 % of them reach the target and 14.04% abort.

TIME STRIKE assumes an aircraft that suffers a weather abort is still subject to attrition; in other words, the aircraft goes all the way to the target before discovering it can't deliver the weapons. TIME STRIKE also assumes the mission doesn't hit a an alternate (dump) target on a weather abort, and the munitions aren't consumed unless the aircraft is lost. This is reasonable, because the cost and availability of modern smart munitions makes an aircrew reluctant to waste them on a dump target. However, carrier-based aircraft do dump munitions due the dangers of an arrested carrier landing while carrying heavy weapons; if TIME STRIKE is ever used for naval forces, we can easily accommodate this change.

F. DATA FILTERS AND OPERATIONAL LIMITS

TIME STRIKE uses a number of factors outside of the formulation to limit the number of alternate sortie types. This is necessary because the number of possible combinations is very large. A typical scenario may contain 9 aircraft types, 90 target types, 300 delivery profiles, and 60 weapon types; in addition, aircraft may carry smaller loadouts of the same weapon to extend their range. When combined with multiple time periods, these combinations can easily lead to an LP containing several hundred thousand variables.

The first step in TIME STRIKE preprocessing is to remove *dominated profiles* from the database. These are delivery profiles for a particular aircraft-weapon-target combination that have a lower effectiveness and a higher attrition than another available profile in that weather state. This simple screen removes up to 30% of the possible aircraft-weapon-profile combinations.

The second step is adopted from HEAVY ATTACK, and involves removing operationally infeasible combinations of aircraft, weapons, and targets from the database. This is done externally, and the amount of reduction depends on how many cases the user is willing to rule out.

Next, we filter the inputs based on two user-supplied settings: the minimum expected kills per sortie (EKS); and the maximum attrition per sortie. Attrition and EKS limits are present in various forms in the existing models' preprocessors, but we have emphasized them in TIME STRIKE. An aircraft-weapon-delivery profile combination that has a probability of .001 of hitting a target and a probability of attrition of .25 is unlikely to be chosen in the optimization, and would *never* be chosen in reality. Therefore, we have urged to users to be aggressive with these filters and throw out as many excess variables as possible prior to running the LP. Computational experience with TAM shows that LP's in this class only choose a few hundred deliveries out of several hundred thousand, so it makes sense to remove the inefficient alternatives before presenting them to the model.

The final screen is based on an operational constraint that is not treated in the existing models: the minimum operating altitude in the period, commonly known as the *hard deck*. Hard decks are real and crucial operating constraints in modern air warfare. If the theater commander decides to fight a medium-altitude war such as DESERT STORM, a great number of delivery tactics are simply not available. In addition, weapons effectiveness, particularly for visual deliveries, varies greatly with release altitude.

HEAVY ATTACK's average weapon scheme picks the best delivery in each weather state and computes a weighted effectiveness; however, the utility currently in use does not consider delivery altitudes and may include invalid combinations. The TAM and MIXMASTER preprocessors choose the best profile for each aircraft-weapon-weather state combination, but some of these profiles may be invalid as well. The latter scheme is also inefficient; if the same profile is the best for each weather state, TAM and MIXMASTER will use 6 decision variables to represent this single alternative. TIME STRIKE's explicit use of the delivery profile allows filtering on the hard deck and prevents duplicate representation of the same sortie type in the model.

Figure 10 shows a typical reduction due to applying these filters. Reductions of an order of magnitude in the number of sortie cases are not uncommon.

	NUMBER OF CASES
INITIAL DATABASE	37,000
after excluding DOMINATED PROFILES	26,000
after excluding OPERATIONALLY INFEASIBLE CASES	10,000
after applying FILTERS (effectiveness, attrition, hard decks)	3,000

Figure 10. Data filtering for TIME STRIKE. Applying filters for dominated profiles, operational infeasibility, effectiveness, attrition, and hard deck settings can remove over 90% of the possible sortie combinations prior to running the model. Using the filters can drastically reduce the size of the LP.

G. TWO-THEATER MODELING

Currently, the US national military strategy requires support of two near-simultaneous "major regional conflicts" (MRC's). Unfortunately, none of the existing models allow for two theaters.

TIME STRIKE allows for two-theater campaigns, so the analyst can develop requirements for both theaters simultaneously. The first campaign starts in the first time period, and the second campaign can start in any time period. The analyst can divide the budgets among the theaters or use additional constraints to bound the overall resource consumption in both theaters.

Another important capability in TIME STRIKE is the ability to *swing* aircraft from the first campaign to the second. Force reductions have led the USAF to adopt a swing doctrine for certain high-value, high-leverage assets such as the F-117. However, the question of when to swing these aircraft and how many to swing is an open issue. TIME STRIKE can optimize the timing and number of swing aircraft, given user-supplied bounds on the number that can swing and when they can swing.

All the machinery available in one campaign in TIME STRIKE is implemented in the two-theater formulation. The theaters have separate target sets, separate weather distributions, BDA rates, regeneration rates, sortie rates, force structures, and so on. This capability does not come without cost; a two-theater LP can become very cumbersome, making intelligent use of the filters very important.

H. BUDGET CONSTRAINTS

HEAVY ATTACK and MIXMASTER do not contain budget constraints. The most common version of TAM has one budget constraint, which is applied globally. Aircraft and weapons in TAM have marginal costs, and the model can either purchase assets subject to a spending constraint or minimize the amount spent to achieve certain goals.

The TAM budget scheme has two shortcomings. First, the single budget isn't flexible enough to account for different types of resources consumed by weapons and aircraft, such as procurement funds,

airlift, and airfield space. Second, it doesn't distinguish expenditures on aircraft, which are long-term assets, and munitions, which are expendables.

TIME STRIKE contains four different budgets in two categories, and the analyst can use any or all of them as constraints. The categories are called *carry* and *no-carry* to denote how the resource can be spent across time. A carry budget represents a resource such as procurement funds; it has no relation to time within the model, because we are trying to determine the investment necessary to meet campaign goals in a future conflict. Conversely, a no-carry budget represents a resource that must be used within a time period; unused resources don't "carry" to succeeding periods. This budget models resources such as airlift, which must be spent when available and can't be saved. There are two carry and two no-carry budgets available in TIME STRIKE.

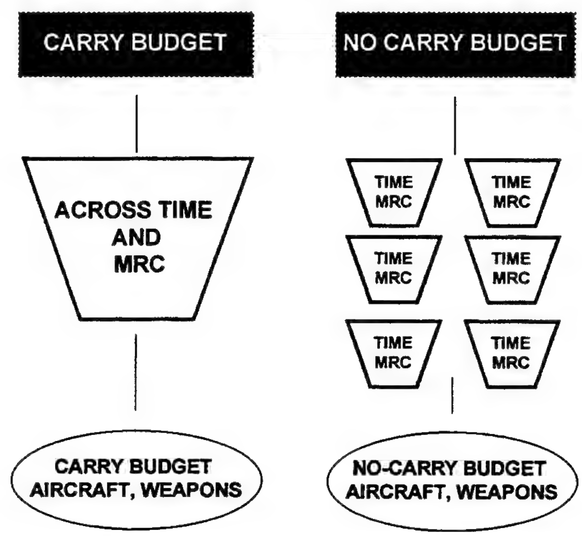

Figure 11. TIME STRIKE budgets. Each budget type has its own pool of aircraft and weapons. The "carry" budgets represent resources that can be spent across time and theater (MRC), while the no-carry budgets represent resources that can only be spent when and where they are available.

One limitation of TIME STRIKE is that assets can only be bought in one budget; purchased assets do not consume a vector of resources, as shown in Figure 11. For example, buying a weapon cannot simultaneously consume procurement dollars and mobility resources; the assets are only available in each budget, and each budget must have its own upper bounds on aircraft and weapon purchases. This may seem to be an unreasonable assumption, but we chose to implement budgets this way to avoid unnecessarily complicating the formulation to address a set of problems that have yet to come up in practice.

I. AIRCRAFT ATTRITION

TIME STRIKE uses an approach similar to TAM's for modeling aircraft attrition. Each feasible combination of aircraft, weapon, target, and delivery profile suffers an input proportion of attrition based on the time period; however, targets killed by the model do not affect these attrition rates. Changes in attrition rates due to enemy air-to-air or surface-to-air assets as a function of time are determined externally to the model. TIME STRIKE uses these inputs to constrain or minimize attrition, depending on how the user is running the model.

However, the user has the option in TIME STRIKE to specify how attrition affects sortie generation. In TAM, attrition reduces the number of available sorties. TIME STRIKE offers this option (see Appendix C), but also offers the option of turning off the sortie reduction. The first case is the same as assuming no replacement aircraft are available; the second case assumes adequate replacement aircraft are available immediately, so no sorties are lost due to attrition. However, TIME STRIKE can still constrain aircraft losses in the second case.

J. IMPLEMENTATION

TIME STRIKE and QUICK STRIKE are straightforward LP's that do not require special solution techniques. Therefore, the Air Force has chosen to use GAMS (Brooke, Kendrick, and Meeraus [1992]) to generate the models. The system can use any commercial LP solver that interfaces with GAMS, and input and output are managed by a graphical user interface that is currently under development. We expect users to run QUICK STRIKE and TIME STRIKE on a variety of hardware platforms.

III. MODEL FORMULATION

A. INDICIES

These are the indices used by the model.

i	aircraft	
j	weapon	
k	target	
l	loadout	
p	delivery profile	
t	time period	
w	weather state	
d	distance band	
c	target class	
$c1 \subseteq c$		target classes with phase goals
$c2 \subseteq c$		target classes without phase goals
f	weapon component family	
q	weapons qualification family	
h	phase	
b	budget	
$b1 \subseteq b$		no-carry budgets
$b2 \subseteq b$		carry budgets
m	major regional conflict (MRC) or theater	

We also use the following to denote valid n-tuples (correspondences) of the index arguments. For example, **cc(k,c)** denotes the set of all admissible target-target class combinations.

cc(k,c)	target-target class correspondence
fc(j,f)	weapon-component family correspondence
qc(j,q)	weapon-qualification family correspondence
r(i,j,l,d)	aircraft-weapon-loadout-distance band correspondence
wc(i,j,p,w)	aircraft-weapon-profile-weather state correspondence

hc(c,k,h) target class-target-phase correspondence

The following correspondence is included to clarify the documentation:

mc(c,k) correspondence of target **k** with the target class **c** containing the highest goal for target **k**

We define these n-tuples with considerable care to limit the formulation to a manageable number of combinations.

B. DATA

The following are the data used in TIME STRIKE:

$ACCOSTS_{ib}$	budget **b** resource consumed per aircraft **i**
$ACMAXBUY_{ib}$	maximum number of aircraft **i** available for purchase in budget **b**
$ATTR_{mijkpt}$	losses per sortie for combination **i,j,k,p** in time period **t** in MRC **m**
ATTRWGT	objective function weight for attrition
$BDAREG_{mkt't}$	expected number of targets **k** dead or in repair in time period **t** that were originally struck in time period **t'**, in MRC **m** (derived in Appendix A)
$BDGLIMITS_{m,b2}$	resource limit for budget **b2** in MRC **m**
$BDGLIMITT_{m,b1,t}$	resource limit for budget **b1** in period **t** in MRC **m**
$BUYWGT_{b}$	objective function weight for spending in budget **b**
$CUMARRIVE_{mjt}$	number of weapon **j** scheduled to arrive in period **t** in MRC **m**
$ENDDAY_{mt}$	last day of period **t** in MRC **m**, relative to the start of MRC **m**
$EKS_{mijklpt}$	expected kills per sortie for aircraft **i**, weapon **j**, target **k**, loadout **l**, profile **p**, and time period **t** in MRC **m**
$FAMLIM_{mf}$	maximum number of common components available for weapon family **f** in MRC **m**
$GOAL_{mct}$	proportion of targets in target class **c** to be killed to achieve the goal in time period **t** in MRC **m**
GOALWGT	objective function weight for goal achievement
$HISTFORECAST_{mw}$	cumulative proportion of forecasts for weather states **1** through **w** in MRC **m**

INVENT_{mj}	inventory of weapon **j** on-hand in MRC **m**
KFORCE_{mijkt}	number of target **k** that must be killed in MRC **m** by aircraft **i** with weapon **j** in time period **t**
LOAD_{l}	number of weapons carried per sortie for loadout **l**
MAXLOSS_{mi}	maximum losses of aircraft **i** allowed in MRC **m**
MUNWGT	objective function weight for munitions use
NABORT_{mijp}	proportion of sorties by aircraft **i** flying profile **p** with weapon **j** not aborted inflight in MRC **m**
NDAYS_{t}	number of days in time period **t**
NPHASEPEN_{m}	objective function reward for killing targets without phase goals in MRC **m**
PGOALS_{mch}	proportion of targets in class **c** to be killed in MRC **m** to achieve phase goal **h**
PHASEPEN_{mh}	objective function penalty per day for not meeting phase goal **h** in MRC **m**
PPEN_{mct}	objective function penalty for not meeting the time-scripted goal for target class **c** by the end of period **t** in MRC **m**
PPENALTY_{mch}	importance value of target class **c** in phase **h** in MRC **m**
PROPORTION_{iq}	proportion of aircrews manning aircraft **i** qualified to drop weapons in qualification class **q**
PTGOAL_{mhk}	maximum PGOALS_{mch} for all target classes containing target **k**
SORTSWITCH	binary input that determines whether or not attrition reduces available sorties (1=yes, 0=no)
SORTWGT	objective function weight for sorties
SR_{mit}	sorties per day for aircraft **i** in period **t** in MRC **m**
SWINGS_{it}	maximum number of aircraft **i** allowed to swing in period **t**
TBLIMITS_{b}	total spending limit for budget **b**
THRESHOLD_{mh}	proportion of goal **h** required in MRC **m** before next phase can start
TIMEAC_{mit}	number of aircraft **i** scheduled to arrive by period **t** in MRC **m**
TOTTGTS_{mkd}	total number of type **k** targets in distance band **d** in MRC **m**

$TOTTGT_{mk}$	total number of type **k** targets in MRC **m**
$TSORT_{mijkpt}$	expected number of sorties per aircraft for combination **i, j, k, p** in period **t** in MRC **m**, including attrition
TV_{mkt}	target value for target **k** in period **t** in MRC **m**
TVDWGT	objective function weight for TVD
$WPNCOSTS_{jb}$	resources consumed per weapon **j** bought in budget **b**
$WPNMAXBUY_{jb}$	maximum number of weapon **j** available in budget **b**

C. VARIABLES

All variables are given in lower case. The following variables represent the value of the various objective functions:

z	time-scripted goal objective value
a	minimum attrition objective value
co_b	minimum cost objective value for budget **b**
v	target value destroyed objective value
p	phase-goal objective value

These are the decision variables:

$x_{mijklpt}$	sorties assigned
$pdiff_{mkct}$	proportion of kills below goal **c** for target **k** in MRC **m**
$pdiffph_{mht}$	positive difference from goal for phase **h** at the end of time period **t**
$wpnbt_{mjbt}$	weapons of type **j** bought in budget **b** in time period **t** and MRC **m**
$acbt_{mibt}$	aircraft of type **i** bought in budget **b** in time period **t** and MRC **m**
$period_{mht}$	binary variable with value 1 if a switch to phase **h+1** occurs in period **t**
$leaving_{it}$	aircraft **i** swinging from MRC **1** in period **t**
$arriving_{it}$	aircraft **i** swinging to MRC **2** in period **t**
$onhanduse_{mj}$	existing inventory of weapon **j** used in MRC **m**
$pkills_{mckht}$	number of target **k** kills attributed to class **c** and phase **h** by the end of time period **t** in MRC **m**

D. OBJECTIVE FUNCTIONS

As mentioned previously, TIME STRIKE offers five different objective functions. Our experience has shown these LPs often have multiple optimal solutions, so we use weights on attrition, sorties, munitions expenditures, weapons and aircraft purchases, and kills by depth to break these ties. These penalty terms are defined below:

$$at = ATTRWGT * \sum_{mijklpt} ATTR_{mijkpt} * x_{mijklpt}$$

$$so = SORTWGT * \sum_{mijklpt} x_{mijklpt}$$

$$mu = MUNWGT *$$

$$\sum_{mijklpt} LOAD_l * \left[ATTR_{mijkpt} * \left(1 - NABORT_{mijp}\right) + NABORT_{mijp}\right] * x_{mijklpt}$$

$$bu = \sum_{mb}\left[BUYWGT_b * \left(\sum_{it} ACCOSTS_{ib} * acbt_{mibt} + \sum_{jt} WPNCOSTS_{jb} * wpnbt_{mjbt}\right)\right]$$

The following are the objective functions.

1. Minimize the penalties associated with unmet, time-scripted goals:

$$\min z = \sum_{m,(k,c)\in cc(k,c),t} \left(PPEN_{mct} * pdiff_{mkct}\right) + at + so + mu + bu$$

2. Minimize attrition for fixed kill goals:

$$\min a = \sum_{mijklpt} ATTR_{mijkpt} * x_{mijklpt} + so + mu + bu$$

The goals are inelastic in this objective function. If the LP can't achieve the goals, the model is infeasible.

3. Minimize cost in a single budget for fixed kill goals:

$$\min co_b = \left(\sum_{mit} ACCOSTS_{ib} * acbt_{mibt} + \sum_{mjt} WPNCOSTS_{jb} * wpnbt_{mjbt} \right)$$
$$+at + so + mu$$

With this objective, user can minimize one of the four budgets and constrain the other three. The kill goals in this formulation are inelastic.

4. Maximize the weighted sum of TVD and time-scripted goals:

$$\max v =$$
$$TVDWGT * \sum_{mkt} \left[TV_{mkt} * \sum_{ijlp,t' \le t} \left(BDAREG_{mkt't} * EKS_{mijklpt'} * NABORT_{mijp} * x_{mijklpt'} \right) \right]$$
$$+GOALWGT * \sum_{m,(k,c) \in cc(k,c),t} \left(PPEN_{mct} * pdiff_{mkct} \right)$$

This objective function is included in TIME STRIKE to provide a way to optimize using target values. The user can reward or penalize TVD and goal achievement by setting the weights.

5. Minimize the time-weighted penalties associated with unmet phase goals:

$$\min P = \sum_{mht} PHASEPEN_{mh} * ENDDAY_{mt} * pdiffph_{mht} -$$
$$\sum_{\substack{m,i,(k,c2) \in cc(k,c2), \\ j,l,p, \\ t=\max(t)}} NPHASEPEN_m * \left(BDAREG_{mktt} * EKS_{mijklpt} * NABORT_{mijp} * x_{mijklpt} \right)$$

In this objective, goals accrue increasing penalties the longer they remain unsatisfied. Targets not included in the phase goals are consolidated, and the user can either reward or penalize killing them in the objective function. This objective function requires a mixed-integer formulation to force the model to achieve the phases in hierarchical order; see Appendix B for details on TIME STRIKE's solution method.

E. SIMPLE BOUNDS

In TIME STRIKE, all variables are nonnegative. We also use simple upper bounds on all variables; these bounds aren't necessary in most cases, but using them tends to decrease solution time. In addition, most commercial solvers have a "presolve" feature that relies on bounds to identify rows and columns that can be eliminated from the LP.

The limits on sorties are given by the maximum number of aircraft available times the number of days in the period times the sortie rate:

$$x_{mijklpt} \leq SR_{mit} * NDAYS_t * TIMEAC_{mit} * \sum_b ACMAXBUY_{ib}$$

The next two proportions naturally have upper bounds of 1:

$$pdiff_{mkct} \leq 1.0$$
$$pdiffph_{mht} \leq 1.0$$

The numbers of weapons and aircraft bought in a time period are limited by the total purchases allowed:

$$wpnbt_{mjbt} \leq WPNMAXBUY_{jb}$$
$$acbt_{mibt} \leq ACMAXBUY_{ib}$$

The number of aircraft swinging to or from a theater in a period is limited by the total number of swings allowed for that aircraft type up to and including that period:

$$leaving_{it} \leq \sum_{t' \leq t} SWINGS_{it'}$$
$$arriving_{it} \leq \sum_{t' \leq t} SWINGS_{it'}$$

The number of on-hand weapons used is limited by the amount available in the theater:

$$onhanduse_{mj} \leq INVENT_{mj}$$

The final bound concerns the number of kills allocated to a phase goal. In TIME STRIKE, phase goals for a target class must be nondecreasing across phases, so $\mathbf{PGOALS_{mch}}$ is cumulative. For example, if the Phase 1 goal for target class 1 is 20 kills, the Phase 2 goal must be greater than 20. However, the $\mathbf{pkills_{mckht}}$ variables are the actual kills assigned to the phase, so kills allocated to a phase cannot be any higher than the difference between the goal for that phase and the goal for the previous phase:

$$pkills_{mckht} \leq \left(PGOALS_{mch} - PGOALS_{m,c,h-1}\right) * \sum_{d} TOTTGTS_{mkd}$$

F. CONSTRAINTS

The following are the explicit constraints available in the model. The notation $|_{\text{<condition>}}$ following a term means that term is only included if <condition> is true. We use this to identify terms that do not apply due to either user settings or inadmissible values of indices. The numbers in parenthesis following the title indicate which objective functions use the constraint.

1. Elastic goal constraints (1, 4):

$$\frac{\sum_{mijlp,t'\leq t}\left(BDAREG_{mkt't} * EKS_{mijklpt'} * NABORT_{mijp} * x_{mijklpt'}\right)}{GOAL_{mct} * TOTTGT_{mk}} + pdiff_{mkct} = 1.0$$

for all $\mathbf{m}, \mathbf{(k,c)} \in \mathbf{cc(k,c)}, \mathbf{t}$

Note that a target appearing in multiple target classes will have multiple positive $(\mathbf{pdiff_{mkct}})$ differences. This is intentional; we want a target affecting multiple goals to accumulate multiple penalties in the objective function. Two important restrictions in TIME STRIKE are: first, goals for each target do not decrease across time; and second, killing targets beyond the goal is not allowed.

2a, 2b. Aircraft-sortie constraints, and aircraft buy constraints (1, 2, 3, 4, 5):

$$\sum_{jklp} \frac{x_{mijklpt}}{TSORT_{mijkpt}} \le TIMEAC_{mit} + \sum_{t' \le t, b} acbt_{mibt'} - \left. \sum_{jklp, t'<t} ATTR_{mijkpt'} * x_{mijklpt'} \right|_{SORTSWITCH=1} + \sum_{t' \le t} \left(arriving_{it'}\big|_{m=1} - leaving_{it'}\big|_{m=2} \right) \tag{2a}$$

for all **m, i, t**

If **SORTSWITCH** is set to 0, the model's sortie generation isn't affected by attrition, and $\mathbf{TSORT_{mijkpt}}$ is set to $\mathbf{SR_{mit}}$*$\mathbf{NDAYS_t}$ for all periods. Also, the attrition sum in equation (2a) is omitted. Otherwise, we use the form of $\mathbf{TSORT_{mijkpt}}$ derived in Appendix C and include the attrition sum.

The aircraft buy constraints are as follows:

$$\sum_{mt} acbt_{mibt} \le ACMAXBUY_{ib} \tag{2b}$$

for all **i, b**

3. Attrition constraints (1, 3, 4, 5):

$$\sum_{jklpt} ATTR_{mijkpt} * x_{mijklpt} \le MAXLOSS_{mi}$$

for all **m, i**

TIME STRIKE currently constrains aircraft attrition by aircraft type and MRC. Constraining attrition by period and across MRCs is a simple matter and may be added in the future.

4a, 4b, 4c. Weapons use, family, and buy constraints (1, 2, 3, 4, 5):

$$\sum_{iklp,t'\leq t} LOAD_l * \left[ATTR_{mijkpt'} * \left(1 - NABORT_{mijp}\right) + NABORT_{mijp}\right] * x_{mijklpt'}$$

$$\leq$$

$$onhanduse_{mj} + \sum_{jb,t'\leq t} wpnbt_{mjbt'} + CUMARRIVE_{mjt}$$

for all **m, j, t** **(4a)**

$$\sum_{j \in fc(j,f)} onhanduse_{mj} \leq FAMLIM_{mf} \qquad \textbf{(4b)}$$

for all **m, f**

$$\sum_{mt} wpnbt_{mjbt} \leq WPNMAXBUY_{jb} \qquad \textbf{(4c)}$$

for all **j, b**

Constraint (4a) counts the number of sorties that either drop bombs on a target or suffer attrition during an in-flight weather abort; in both cases, the weapons are consumed. Constraint (4b) addresses weapons that share common components, which is an important issue in munitions allocation. $\mathbf{FAMILYLIM_{mf}}$ gives the total number of available components, but this limit only applies to on-hand inventory. TIME STRIKE assumes purchased or arriving weapons are complete rounds.

5a, 5b, 5c. Budget constraints (1, 2, 3, 4, 5):

$$\sum_{i} ACCOSTS_{i,b1} * acbt_{m,i,b1,t} + \sum_{j} WPNCOSTS_{j,b1} * wpnbt_{m,j,b1,t} \leq BDGLIMITT_{m,b1,t}$$

for all **m, b1, t** **(5a)**

$$\sum_{it} ACCOSTS_{i,b2} * acbt_{m,i,b2,t} + \sum_{jt} WPNCOSTS_{j,b2} * wpnbt_{m,j,b2,t} \leq BDGLIMITS_{m,b2}$$

for all **m, b2** **(5b)**

$$\sum_{mit} ACCOSTS_{i,b2} * acbt_{m,i,b2,t} + \sum_{mjt} WPNCOSTS_{j,b2} * wpnbt_{m,j,b2,t} \leq TBLIMITS_{b2}$$

for all **b2** **(5c)**

We track weapons purchases by time period for no-carry (**b2**) budgets as a convenience; indexing these purchases by time period isn't necessary. If the user wants all of a budget available to either MRC, he can set the individual MRC budget limits to the overall limit, making the last budget constraint redundant.

6. Kills by distance constraints (1, 2, 3, 4, 5):

$$\sum_{(i,j,l)\in r(i,j,l,d),p,t'\leq t} \left(BDAREG_{mkt't} * EKS_{mijklpt'} * NABORT_{mijp} * x_{mijklpt'} \right) \leq \sum_{d'\leq d} TOTTGTS_{mkd't}$$

for all **m, k, t,** and **d**

These constraints limit aircraft-weapon-loadout combinations to targets in valid distance bands. They are cumulative to allow longer-range deliveries to kill close-in targets. As a result, the **x** variables do not need an explicit index for distance (as is used in TAM). In addition, targets that do not regenerate only require one constraint per distance band for each MRC; this eliminates a great many redundant constraints and is implemented in the GAMS code.

7. Weather constraints (1, 2, 3, 4, 5):

$$\sum_{(j,p)\in wc(i,j,p,w),kl} x_{mijklpt} \geq HISTFORECAST_{mw} * \sum_{jklp} x_{mijklpt}$$

for all **m, i, w, t**

These constraints force the model to schedule sorties in proportion to the average weather forecast. To maintain feasibility, TIME STRIKE requires a dummy target and a dummy weapon that each aircraft can employ in each weather state. Otherwise, TIME STRIKE would force sorties to be scheduled for aircraft with no valid sortie combinations in particular weather states, making the model infeasible.

Constraints (6) and (7) model the distance and weather constraints that exist in TAM without explicitly indexing the $\mathbf{x}_{mijklpt}$ variables for distance and weather. This means that TIME STRIKE will have far fewer variables than TAM for the same inputs. For example, if we were using 6 weather states and 7 distance bands, 10,000 sortie variables in TIME STRIKE would be equivalent to 420,000 sortie variables in TAM.

8. Weapons qualification constraints (1, 2, 3, 4, 5):

$$\sum_{j \in qc(j,q), klpt} x_{mijklpt} \leq PROPORTION_{iq} * \sum_{jklpt} x_{mijklpt}$$

for all m and (i,q) with $PROPORTION_{i,q} > 0$

These constraints model situations where only a certain proportion of an aircraft's aircrews are qualified to employ a weapon, or only a certain proportion of an aircraft type are equipped to drop a weapon.

9. Elastic phase-goal constraints (5):

$$\frac{\displaystyle\sum_{(k,c1) \in cc(k,c1)} PPENALTY_{m,c1,h} * pkills_{m,c1,k,h,t}}{\displaystyle\sum_{(k,c1) \in hc(c1,k,h)} TOTTGTS_{mk} * PPENALTY_{m,c1,h} * \left(PGOALS_{m,c1,h} - PGOALS_{m,c1,h-1} \Big|_{h>1} \right)} + pdiffph_{mht} = 1.0$$

for all **m, h, t**

This constraint measures the *weighted* proportional difference between the phase goal and number of kills in time period **t**, and sums only over targets that have goals within a particular phase. The penalties give the user influence over what targets are killed first within a phase.

10. Phase-start constraints (5):

$$\sum_{c1 \in mc(k,c1),h} pkills_{m,c1,k,h,t} \leq TOTTGT_{mk} * PTGOAL_{m,h=1,k} +$$

$$TOTTGT_{mk} * \left[\sum_{t' \leq t, h < \max(h)} \left(PTGOAL_{m,h+1,k} - PTGOAL_{mhk} \right) * period_{mht'} \right]$$

for all **m, k, t**

This constraint enforces starting times for phases. Targets in a particular phase goal cannot be killed until $\mathbf{period_{mht}}$ becomes 1, which only occurs if the preceding phase has met its threshold. Since targets can appear in multiple classes, this constraint is only enforced for the target class that has the largest goal for the target. Constraint (13) constrains kills assigned to other classes containing the target.

The expression on the right-hand side is somewhat clumsy because the phase goal data is presented to the model in cumulative terms, but the model needs to know the incremental difference in the goal from one phase to the next. We have chosen to document this constraint in this way to make the math match the form of the input data.

11a, 11b, 11c, 11d, 11e. Phase-preemption constraints (5):

$$period_{mht} \leq (1 - pdiffph_{mht}) / THRESHOLD_{mh} \qquad \textbf{(11a)}$$

for all **m**, **h** < max(**h**), **t**

This constraint induces preemption among phases. Variable $\mathbf{period_{mht}}$ prevents the model from killing targets in succeeding phases by staying at 0 until $\mathbf{pdiffph_{mht}}$ gets small enough to allow the model to switch. $\mathbf{THRESHOLD_{mh}}$ allows for flexibility by defining what proportion of the weighted kills must be achieved before the model can proceed to the next phase.

The notion of a threshold is an important one. If TIME STRIKE had no threshold, the phases would be totally preemptive and failure to kill one difficult target could stop the entire campaign. On the other hand, the difficult targets aren't forgotten, because the penalties associated with positive values of $\mathbf{pdiffph_{mht}}$ increase across time.

In the LP relaxation, $\mathbf{period_{mht}}$ is a continuous variable. The following constraints stop the model from hitting phase targets before the previous phase's threshold is met, and force the continuous relaxation of the $\mathbf{period_{mht}}$ variables closer to binary values.

$$\sum_{t' \le t} period_{m,h-1,t'} \ge (1 - pdiffph_{mht}) \qquad \textbf{(11b)}$$

for all **m, h** > 1, **t**

$\mathbf{period_{mht}}$ can be 1 only in the period in which the goal meets or exceeds its threshold. The following set-packing constraints ensure this:

$$\sum_{t} period_{mht} \le 1 \qquad \textbf{(11c)}$$

for all **m, h**

The next set of constraints prevent a phase from becoming active before the prior phase has met its threshold. In addition, these constraints "sharpen" the LP constraint space to encourage continuous solutions with binary $\mathbf{period_{mht}}$ values.

$$period_{m,h+1,t} \le \sum_{t' \le t} period_{mht'} \qquad \textbf{(11d)}$$

for all **m, h** < max(**h**)-1, **t**

TIME STRIKE requires that achievement in a phase must be nondecreasing across time. Experience has shown the objective function penalties tend to promote this behavior, but we use explicit constraints to ensure it occurs:

$$pdiffph_{mht} \ge pdiffph_{m,h,t+1} \qquad \textbf{(11e)}$$

for all **m, h, t** < max(**t**)

Again, the intent of this entire group of constraints is to ensure proper phase switching and to get the model to force the continuous relaxation of the binary $\mathbf{period_{mht}}$ variables as close as possible to values

of 0 or 1. As discussed in Appendix B, TIME STRIKE uses a sequence of LP solves to fix these variables, rather than resorting to branch-and-bound.

12. Kill-assignment constraints (5):

$$\sum_{c1 \in mc(k,c1),h} pkills_{m,c1,k,h,t} = \sum_{mijlp,t' \leq t} \left(BDAREG_{mkt't} * EKS_{mijklpt'} * NABORT_{mijp} * x_{mijklpt'} \right)$$

for all **m, k, t**

It is necessary to partition the targets that are dead or in repair among target classes and phases to compute correct values for the $\mathbf{pdiffph_{mht}}$ variables. Since targets can be members of multiple classes, TIME STRIKE only constrains the class having the maximum goal for the target.

13. Multiple-class kill constraints (5):

$$pkills_{m,c1,k,h,t} \leq pkills_{m,c \in mc(k,c1),k,h,t}$$

for all **m, (k,c1)∈cc(k,c1), h, t**

These constraints allocate target kills for targets that are members of multiple classes. Constraint (12) only involves the target class with the maximum goal for each target in a phase, and that target class appears on the right-hand side of these constraints. The other target classes containing the target are constrained at either their simple upper bound or the value of the maximum-goal class. As an example, consider a target **k** in phase **h** and period **t** with a goal of 8 in Class 1, 10 in Class 2, and 13 in Class 3. Constraint (12) will divide (or constrain) the number of kills so that $\mathbf{pkills_{m,3,k,h,t}}$ is less than or equal to 13. (13) will constrain $\mathbf{pkills_{m,1,k,h,t}}$ to be less than $\mathbf{pkills_{m,3,k,h,t}}$, and will also constrain $\mathbf{pkills_{m,2,k,h,t}}$ to be less than $\mathbf{pkills_{m,3,k,h,t}}$. If $\mathbf{pkills_{m,3,k,h,t}}$ is less than 8, the other two variables will be set to 8; if it is 11, then $\mathbf{pkills_{m,1,k,h,t}}$ and $\mathbf{pkills_{m,2,k,h,t}}$ will be set to their upper bounds of 8 and 10, respectively.

14. Phase-kill importance constraints (5):

$$\frac{\sum_{c1 \in cc(k,c1)} pkills_{m,c1,k,h,t}}{\sum_{c1 \in cc(k,c1)} \left(PGOALS_{m,c1,h,k} - PGOALS_{m,c1,h-1,k} \Big|_{h>1} \right)} \geq \frac{\sum_{c1 \in cc(k,c1)} pkills_{m,c1,k,h+1,t}}{\sum_{c1 \in cc(k,c1)} \left(PGOALS_{m,c,h+1,k} \Big|_{h<\max(h)} - PGOALS_{m,c1,h,k} \right)}$$

for all **m, (k,c1)** $\in$**cc(k,c1)** with increasing $\mathbf{PPENALTY_{mch}}$ over the phases, **h, t**

Constraints (13) and (14) don't fully account for the case of a target whose importance is low in an early phase but increases in later phases. For example, if Phase 1 hasn't met its threshold constraint, (12b) will prevent any kills against Phase 2 targets. Once the threshold is met the targets can be killed, but the kills may be allocated to Phase 2 instead of Phase 1.

This is a functional as well as an analytical problem. Due to the nature of the penalties, it would take a large number of binary variables to enforce assignment of kills to the proper phases. Consider a target with a Phase 1 requirement of 60 kills and a Phase 2 requirement of 30 kills. Ordinarily, we would assign the first 60 kills to Phase 1 and any remaining kills to Phase 2. This will happen as long as the penalties for this target don't increase across phases; if they do, the allocation may be different than what we intend.

But what does it mean if the penalty in Phase 1 is low and penalty in Phase 2 is high? Perhaps we *want* the kills allocated to Phase 2 because this target wasn't that important to Phase 1. Was it more important to kill the 30 specific targets in Phase 2, or to achieve 90 total kills by the end of Phase 2?

Rather than build in a great deal of model structure to satisfy an ill-defined (and rare) issue, we compromise with these constraints which force the proportion of kills allocated to phases to be nonincreasing. If, in the example above, we had 70 total kills, the constraint forces at least 46.666 of them would have to be allocated to Phase 1 and at most 23.333 of them to Phase 2. This would give both phases the identical proportional achievement of 70/90 = 77.777%.

15a, 15b: Swing aircraft constraints (1, 2, 3, 4, 5):

$$leaving_{it} = arriving_{it} \qquad \textbf{(15a)}$$

for all **i, t**

and

$$\sum_{t' \leq t} arriving_{it'} \leq \sum_{t' \leq t} SWINGS_{it'} \qquad \textbf{(15b)}$$

for all **i, t**

16: Forced-kill constraints (1, 2, 3, 4, 5):

$$\sum_{pl} \left(BDAREG_{mktt} * EKS_{mijkplt} * NABORT_{mijp} * x_{mijklpt} \right) \geq KFORCE_{mijkt}$$

for all **m, i, j, k, t** with $\mathbf{KFORCE_{mijkt}} > 0$

These constraints direct the model to use particular weapons against particular targets at particular times. They are designed to force allocations that the model would not ordinarily make to satisfy requirements exterior to the model; for example, a user may want to test a policy that a particular target *must* be attacked with a particular combination of aircraft and weapon. These constraints are inelastic and may cause the LP to terminate as infeasible.

IV. COMPUTATIONAL EXPERIENCE AND CONTINUED RESEARCH

TIME STRIKE's performance depends on the size of the data set and the objective function used. Current 1-MRC scenarios consider approximately 9 aircraft types, 60 weapons types, 70 target types distributed in 4 distance bands, 7 time periods, 6 weather states, 10 target classes, and 3 phase goals. These problems result in formulations containing approximately 20,000 variables and 7,000 constraints; however, subsequent filtering after one or two tuning runs reduces the LPs to roughly 7,000 variables and 2,000 constraints for time-scripted goals, and 8,000 variables and 3,000 constraints for phase goals. Solution times on current PC's range from 2 to 11 minutes, depending on the choice of LP solver. GAMS overhead in generating the model is modest, ranging from 1 to 3 minutes.

Problems with 2 MRCs can create difficulty on the PC platform unless the system is equipped with more than 32MB of memory. The resulting problems (after filtering) contain 15,000-20,000 variables and up to 7,000 constraints. Nonetheless, we have solved 2-MRC problems with phase goals on a 120-MhZ Pentium™ system with 64MB of memory less than 18 minutes.

Our current testing with analysts at the HQ ACC has shown it is better to run TIME STRIKE a few times with a small number of time periods to identify clearly unproductive sortie combinations and test the feasibility of the campaign goals. Subsequent runs with filtering go considerably faster, and some of the solvers allow us to save previous solutions and do a "warm start" for runs with minor changes.

TIME STRIKE and QUICK STRIKE can also work together. QUICK STRIKE's myopia makes it insensitive to the number of time periods, so a user can run QUICK STRIKE to determine how to set the time periods in TIME STRIKE. Also, TIME STRIKE can give advice to QUICK STRIKE on how to allocate resources such as budgets and attrition across time.

Development will continue with TIME STRIKE. We are building a submodel for the air-to-air part of the campaign, as well as allowing the model to allocate "suppression of enemy air defenses" (SEAD) assets. However, both air-to-air and SEAD modeling lead to an issue that has plagued models of this class, which is how to handle attrition.

Ideally, attrition should be a function of the enemy's order of battle, so as the model destroys enemy surface-to-air and air-to-air assets, attrition should decrease. Unfortunately, adding these effects

directly to an explicit-time model makes it nonlinear and nonconvex—in other words, very difficult to solve. As a result, modelers have avoided putting attrition directly into the model. This is an area where QUICK STRIKE has an advantage over TIME STRIKE, because QUICK STRIKE can recompute attrition rates outside of each myopic solve and send those rates to the next time period. However, QUICK STRIKE's myopia will not guarantee a global solution of when and to what degree enemy air defenses should be attacked.

Other submodels can be improved as well. BDA is an issue in many models used in the Department of Defense, and TIME STRIKE is no exception. TIME STRIKE simplifies and extends current approaches, but is by no means the definitive answer. As we discuss in Appendix A, target regeneration and BDA could be modeled in a comprehensive stochastic model describing the states of the targets in each period, which would strengthen the accounting of the number of targets alive, dead, and awaiting assessment. In addition, the weather submodel deserves more scrutiny. TIME STRIKE uses expected weather and forecast data without regard to the relationships across time among weather states. Also, TIME STRIKE is very conservative on in-flight weather aborts; in actuality, some of these sorties can be redirected to other targets enroute.

Finally, the largest shortcoming of this model is its lack of an intelligent enemy. Optimization brings many advantages to the munitions problem, but so far no one has been able to formulate an LP of this class that allows the enemy to react. We hope that the air-to-air and SEAD submodels currently in work will allow the enemy to make decisions on the allocation and use of his air-to-air and air defense assets, but this will be a difficult task.

Nonetheless, we see optimization as a valuable and appropriate tool for the munitions problem. Optimization can consider a large number of force mixes far faster than a simulation-oriented approach, and LPs such as TIME STRIKE can make many, many allocation and budget decisions in a single run. This, along with the sensitivity analysis available in linear programming, has made these tools invaluable to the US Air Force for the last 25 years. We expect this to be true for the next 25 years as well.

APPENDIX A: TARGET REGENERATION AND BDA EQUATIONS

The TIME STRIKE formulation requires knowing the proportion of targets killed within a time period that regenerate both within the period and in succeeding periods. These proportions are used in the model to keep track of targets that are repaired. This is important, because the goal-oriented objectives operate in terms of targets currently dead, not in terms of the total number of kills.

Determining these proportions is difficult within an optimization, because the repair times, the number of targets repaired, and the number of targets with correct BDA are random variables. In addition, we require that the model wait one planning cycle before detecting a target has regenerated, so some proportion of kills from a period will not be detected until the next period. Finally, the reader should bear in mind that we have to keep the linear program linear, so we have not taken the more rigorous route of developing a comprehensive stochastic model.

We make the following assumptions about target regeneration and BDA:

- Each target of a particular type has a stationary probability of being repairable that is independent of the number of targets killed and the number of times the target has been killed.
- Each target in repair has a independent probability $\mathbf{P_c}$ of regenerating in one planning cycle.
- Targets killed and regenerated require one planning cycle (denoted by **C**) to be detected.
- The number of kills in a time period is a nonrandom quantity determined by the LP, and kills occur uniformly across the time period.
- Each target type has a fixed probability of getting correct BDA. All targets that are dead or in repair with incorrect BDA are restruck in the next planning cycle.
- The probability of correct BDA is independent of whether the target is alive, dead, or in repair, and it is also independent of the number of times the target has been struck.

At the end of each period, we need to know the expected number of targets in each of three possible states: alive and targetable; killed and unrepairable (dead forever); and in repair. However, the

problem is complicated by the fact that a target can be killed, regenerated, and assessed multiple times *within* a period.

We will use the following notation:

$\mathbf{T_0}$	start of the time period
$\mathbf{T_1}$	end of the time period
C	length of a planning cycle
n	number of planning cycles of length **C** in $[\mathbf{T_0}, \mathbf{T_1}]$
$\mathbf{L_i}$	expected number of targets detected as alive at the beginning of the ith planning cycle, $\mathbf{1 \le i \le n}$
$\mathbf{L_0}$	initial number of live targets
$\mathbf{R_i}$	expected number of targets in repair at the beginning of the ith planning cycle, $\mathbf{1 \le i \le n}$
$\mathbf{R_0}$	initial number of targets in repair
$\mathbf{D_0}$	initial number of dead targets
$\mathbf{D_i}$	expected number of dead targets at the at the beginning of the ith planning cycle, $\mathbf{1 \le i \le n}$
K	total number of (nonrandom) kills allocated by the LP in the period, including restrikes
$\mathbf{K_i}$	total number of (nonrandom) kills allocated by the LP in planning cycle **i**, including restrikes
$\mathbf{K_0}$	number of kills allocated by the LP in the last planning cycle of the previous period
$\mathbf{P_r}$	probability the target is repairable after a strike
$\mathbf{NP_r}$	probability a target isn't repairable; $\mathbf{NP_r=1-P_r}$
$\mathbf{P_c}$	probability a target regenerates in the next planning cycle
$\mathbf{NP_c}$	probability a target doesn't regenerate in the next planning cycle; $\mathbf{NP_c=1-P_c}$
B	probability of correct BDA for a target
NB	probability of incorrect BDA for a target; $\mathbf{NB = 1- B}$
r	repair rate for targets that are repaired

Since kills occur uniformly in the period, $\mathbf{K_i = K/n}$ kills occur in each planning cycle. We will also make the following two simplifying assumptions:

- All kills that occur in cycle **i** occur instantaneously at the start of the cycle. Therefore, $\mathbf{K_i = K/n}$ kills occur at the times $\mathbf{T_0}$, $\mathbf{T_0 + C}$, $\mathbf{T_0 + 2C}$, ... $\mathbf{T_O + (n\text{-}1)C}$.

- Targets dead or in repair are correctly assessed after a restrike. In other words, a restrike exposes the true state of the target.

The first assumption is conservative, because it gives the targets credit for regenerating faster than they would if kills followed a time distribution within the cycle. However, planning cycles tend to be short with respect to the length of the time period, so the effect of this assumption is small. The second assumption simplifies the model and is probably realistic. Note that a target *can* be assessed incorrectly after every regeneration; we only assume that there are no bad assessments against restruck targets.

We will begin with a difference equation for the expected number of dead targets. At the beginning of each planning cycle, the model allocates $\mathbf{K_i = K/n}$ kills. However, we expect a proportion of those kills to be restrikes against targets struck in the previous period that have incorrect BDA. Assuming that no target is assessed incorrectly for two planning cycles in a row, we have the following expected number of kills against live targets:

$$K_i - K_{i-1} * NB$$

Of these, we expect a fixed proportion to be unrepairable, so the expected number of dead targets increases by

$$(K_i - K_{i-1} * NB) * NP_r$$

However, some of the restrikes will also result in dead targets. Restrikes go against either dead targets or targets in repair, and restriking a dead target will not change its status. However, a restrike against a target in repair can make the target unrepairable. Since we assume BDA probabilities are independent of repair probabilities, we expect $\mathbf{P_r}$ proportion of the targets we restrike to be in repair. Therefore, the following is the number of dead targets due to restrikes:

$$(K_{i-1} * NB * P_r) * NP_r$$

This leads to the following difference equation:

$$
\begin{aligned}
D_i &= D_{i-1} + \left(K_i - K_{i-1} * NB\right) * NP_r + \left(K_{i-1} * NB * P_r\right) * NP_r \\
&= D_{i-1} + \left[K_i - K_{i-1} * NB * \left(1 - P_r\right)\right] * NP_r \\
&= D_{i-1} + \left[K_i - K_{i-1} * NB * NP_r\right] * NP_r
\end{aligned}
$$

We are interested in $\mathbf{D_n}$, the total number of dead targets at the end of the time period. Expanding the difference equation leads to the following:

$$
\begin{aligned}
D_1 &= D_0 + \left[K_1 - K_0 * NB * NP_r\right] * NP_r \\
D_2 &= D_1 + \left[K_2 - K_1 * NB * NP_r\right] * NP_r \\
&= D_0 + K_1 * NP_r - K_0 * NB * NP_r^{\,2} + K_2 * NP_r - K_1 * NB * NP_r^{\,2} \\
&= D_0 - K_0 * NB * NP_r^{\,2} + \left(1 - NB * NP_r\right) * NP_r * K_1 + K_2 * NP_r \\
&\vdots \\
D_n &= D_0 - K_0 * NB * NP_r^{\,2} + \left[\left(1 - NB * NP_r\right) * NP_r * \sum_{i=1}^{n-1} K_i\right] + K_n * NP_r
\end{aligned}
$$

Since $\mathbf{K_i = K/n}$,

$$
\begin{aligned}
D_n &= D_0 - K_0 * NB * NP_r^{\,2} + \left(1 - NB * NP_r\right) * NP_r * \frac{(n-1) * K}{n} + \frac{K * NP_r}{n} \\
&= D_0 - K_0 * NB * NP_r^{\,2} + K * NP_r - K * NB * NP_r^{\,2} + \frac{K * NB * NP_r^{\,2}}{n} \\
&= D_0 - K_0 * NB * NP_r^{\,2} + K * NP_r - K * NB * NP_r^{\,2} * \left(1 - \frac{1}{n}\right) \\
&= D_0 - K_0 * NB * NP_r^{\,2} + K * NP_r * \left[1 - NP_r * \left(NB - \frac{NB}{n}\right)\right]
\end{aligned}
$$

Note that if we have perfect BDA, $\mathbf{NB = 0}$ and

$$
D_n = D_0 + K * NP_r
$$

We will use the same methods to develop difference equations for the expected number of targets in repair. We assume repair times are exponential, which implies the probability a target completes repair in a planning cycle is independent of the length of time it has been in repair. As a result, we only have to count the targets in repair from cycle to cycle.

The expected number of targets continuing in repair through cycle **i** is

$$NP_c * R_{i-1}$$

The expected number of targets put into repair due to strikes against live targets is

$$(K_i - K_{i-1} * NB) * P_r$$

Again, we have to determine the effect of restrikes. The following is the expected number of targets in repair that are restruck and made unrepairable:

$$(K_{i-1} * NB * P_r) * NP_c * NP_r$$

Some of the targets that would have come out of repair in the planning cycle *return* to repair due to the restrikes. The expected number of these is

$$(K_{i-1} * NB * P_r) * P_r * P_c$$

Adding these terms with the appropriate signs yields the following difference equation:

$$\begin{aligned}
R_i &= NP_C * R_{i-1} + (K_i - NB * K_{i-1}) * P_r - \\
&\quad (K_{i-1} * NB * P_r) * NP_c * NP_r + (K_{i-1} * NB * P_r) * P_r * P_c \\
&= NP_C * R_{i-1} + K_i * P_r - K_{i-1} * P_r * [NB * (1 + NP_c * NP_r - P_r * P_c)] \\
&= NP_C * R_{i-1} + K_i * P_r - K_{i-1} * P_r * [NB * (1 + NP_r - P_c)] \\
&\equiv NP_C * R_{i-1} + K_i * P_r - K_{i-1} * P_r * A
\end{aligned}$$

where we let **A** represent the last term in brackets. Expanding each term of this sequence gives the following general form for $\mathbf{R_n}$:

$$R_1 = NP_C * R_0 - K_0 * P_r * A + P_r * K_1$$

$$\begin{aligned} R_2 &= NP_C * R_1 + K_2 * P_r - K_1 * P_r * A \\ &= NP_C{}^2 * R_0 - NP_C * K_0 * P_r * A + NP_C * P_r * K_1 + K_2 * P_r - K_1 * P_r * A \\ &= NP_C{}^2 * R_0 - NP_C * K_0 * P_r * A + P_r * (NP_C - A) * K_1 + K_2 * P_r \\ &\vdots \end{aligned}$$

$$R_n = NP_C{}^n * R_o - NP_C{}^{n-1} * K_0 * P_r * A + P_r * (NP_C - A) * \sum_{i=1}^{n-1} K_i * NP_C{}^{n-(i+1)} + K_n * P_r$$

Using the assumption that $\mathbf{K_i = K/n}$ and applying the identity

$$\sum_{i=1}^{n-1} ax^{i-1} = a * \frac{(1 - x^{n-1})}{1 - x}$$

reduces the sum:

$$\begin{aligned} \sum_{i=1}^{n-1} K_i * NP_C{}^{n-(i+1)} &= \frac{K}{n} * \sum_{i=1}^{n-1} NP_C{}^{n-(i+1)} \\ &= \frac{K}{n} * \sum_{i'=n-1}^{i'=1} NP_C{}^{i'-1} \quad \text{where } i' = n - i \\ &= \frac{K}{n} * \sum_{i'=1}^{n-1} NP_C{}^{i'-1} \\ &= \frac{K * (1 - NP_C{}^{n-1})}{n * (1 - NP_C)} \\ &= \frac{K * (1 - NP_C{}^{n-1})}{n * P_C} \end{aligned}$$

This leads to the final result:

$$R_n = NP_C{}^n * R_o - NP_C{}^{n-1} * K_0 * P_r * A + \frac{K * P_r}{n} * \left[\frac{(NP_C - A) * (1 - NP_C{}^{n-1})}{P_C} + 1 \right]$$

The expected number of live targets at the end of the period (L_n) is computed from R_n and D_n:

$$L_n = L_0 + R_0 + D_0 - R_n - D_n$$

It is important to note that L_n is the number of targets that are *detected* as alive at the end of the time period. Targets killed in the period and regenerating into the last planning cycle are not detected by the model as being alive, so the model can allocate enough kills to reduce L_n to 0. This number of kills is denoted by K_{max}, and is used as a constraint in the model.

We will illustrate the behavior of K_{max} for a few limiting cases. First, let $P_r = 0$, so nothing can be repaired. This means that $R_0 = 0$ and $R_n = 0$ (which is easily verified by inspection). In this case, we determine K_{max} as follows:

$$\begin{aligned} 0 &= L_0 + D_0 - D_n \\ &= L_0 + D_0 - D_0 + K_0 * NB * 1 - K_{\max} * 1 * \left[1 - 1 * \left(NB - \frac{NB}{n}\right)\right] \end{aligned}$$

$$K_{\max} = \frac{L_0 + K_0 * NB}{1 - NB + \dfrac{NB}{n}}$$

If we have perfect BDA, the above expression reduces to $K_{max} = L_0$. This is what we would expect for no BDA dilution and no target repair.

If the repair rate is infinite, the mean repair time is 0 and targets come back to life instantly. Therefore, $P_c = 1$ and $NP_c = 0$, and R_n is as follows:

$$\begin{aligned} R_n &= \frac{K * P_r}{n} * [1 - A] \\ &= \frac{K * P_r}{n} * [1 - NB * (1 + NP_r - 1)] \\ &= \frac{K * P_r * [1 - NB * NP_r]}{n} \end{aligned}$$

Solve as before for K_{max}:

$$
\begin{aligned}
0 &= L_0 + D_0 - R_n - D_n \\
&= L_0 + D_0 - \frac{K_{\max} * P_r * [1 - NB * NP_r]}{n} - D_0 \\
&\quad + K_0 * NB * NP_r^{\,2} - K_{\max} * NP_r * \left[1 - NP_r * \left(NB - \frac{NB}{n}\right)\right]
\end{aligned}
$$

$$
K_{\max} = \frac{L_0 + K_0 * NB * NP_r^{\,2}}{\dfrac{P_r * [1 - NB * NP_r]}{n} + NP_r * \left[1 - NP_r * \left(NB - \dfrac{NB}{n}\right)\right]}
$$

If we assume perfect BDA and 100% repair (**NB** = 0), then $\mathbf{K_{max}}$ reduces to:

$$
K_{\max} = L_0 * n
$$

This is what we would expect; we kill all the targets in each planning cycle, and they come back instantly and we have to kill them again. If we allow BDA to be less than perfect and assume no repair, then $\mathbf{K_{max}}$ is

$$
K_{\max} = \frac{L_0 + K_0 * NB}{\left[1 - NB * \left(\dfrac{n-1}{n}\right)\right]}
$$

which illustrates the dilution of kills due to incorrect BDA.

Figure A1 shows $\mathbf{K_{max}}$ for varying values of **r** and $\mathbf{P_r}$ in a 30-day time period with **C** = 3, $\mathbf{R_0}$ = 0, $\mathbf{L_0}$ = 100, and **B** = 1.0:

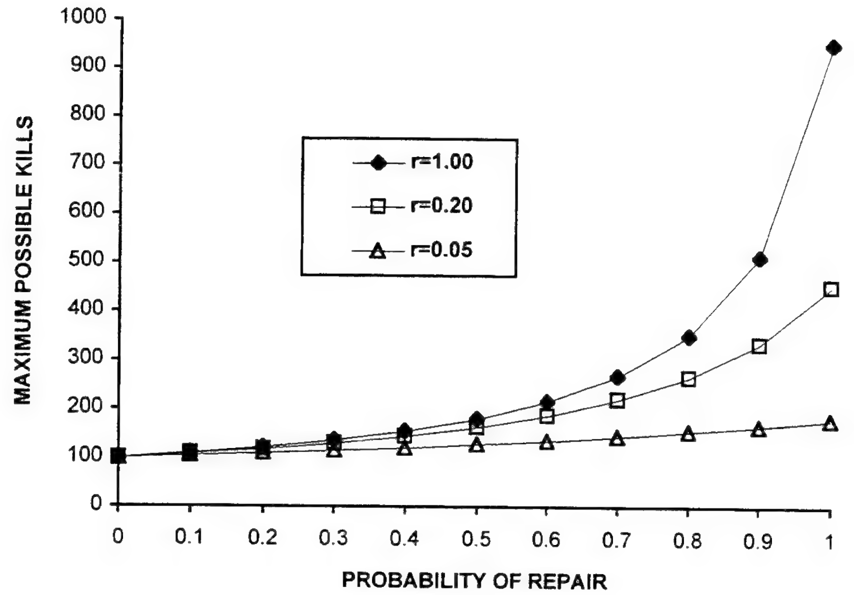

Figure A1. Maximum possible kills (K_{max}) as a function of repair rate (r) and probability of repair. This graph shows the maximum number of kills possible in a 30-day period for various repair rates and probabilities of repair, given that 100 targets are alive at the beginning of the period. Targets with high repair rates and high probabilities of repair can potentially be killed many times within the period. This chart assumes perfect BDA.

As the chart shows, targets with high repair rates and high probabilities of repair can regenerate many times within a time period. If TIME STRIKE did not allow kills against regenerated targets within a period, the model could assume one kill suppresses a target for the entire period, which is very misleading. In Figure A1, the repair rates correspond to mean repair times of 1, 5, and 20 days; if each target has a probability of repair of 1, the model must kill them 9.50, 4.52, and 1.79 times respectively within the period to ensure they are dead at the end of the period.

Figure A2 shows $\mathbf{R}_n$ for the same situation as Figure A1, except that we assume the model has allocated 100 total kills in the period:

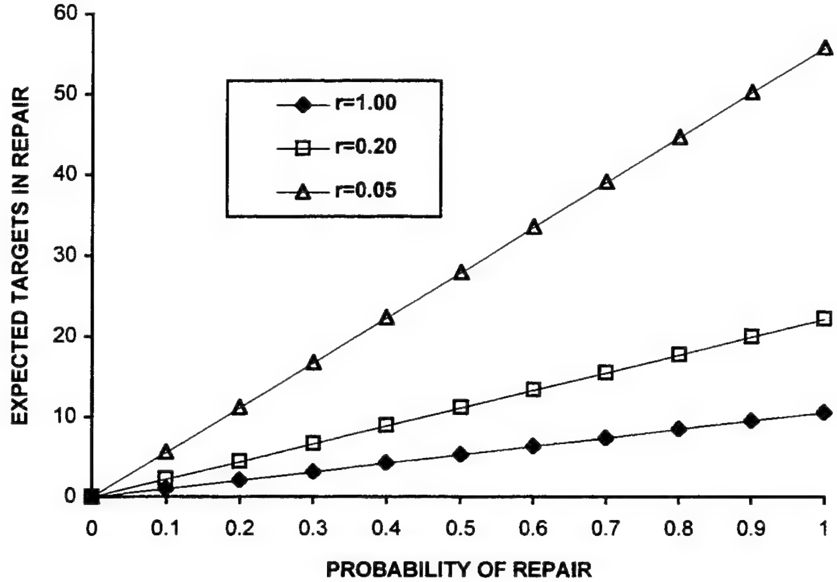

Figure A2. Expected number of targets in repair at the end of the period as a function of repair rate (r) and probability of repair. This graph shows the targets in repair at the end of a 30-day period for various repair rates and probabilities of repair, given that 100 targets are alive at the beginning of the period and the model allocates 100 kills. This chart assumes perfect BDA.

A final point concerns the total number of target regenerations. The version of HEAVY ATTACK currently in use allows targets to regenerate once, while TAM allows infinite regeneration. As it turns out, we can determine the maximum number of regenerations possible in TIME STRIKE. Let TGT denote the total starting number of targets of a certain type and RG denote the total possible number of regenerations for that target type. Then:

$$
\begin{aligned}
RG &= \left(P_r * TGT\right) + \left(P_r * P_r * TGT\right) + \ldots \\
&= TGT * \sum_{i=1}^{\infty} P_r^{\,i} = TGT * \left(\sum_{i=0}^{\infty} P_r^{\,i} - 1\right) = TGT * \left(\frac{1}{1-P_r} - 1\right) \\
&= \frac{TGT * P_r}{1-P_r}
\end{aligned}
$$

Frequently, a user will not know the probability the enemy will repair a struck target. However, he may have information on the enemy's total repair capacity. To limit the total possible number of regenerations implicitly, the analyst can solve for $\mathbf{P_r}$ in terms of RG and TGT and use the result as an input:

$$P_r = \frac{RG}{TGT + RG}$$

This development is a heuristic approach to a complicated stochastic process. In particular, the number of regenerations within a period is a function of the repair rate, the probability of repair, the kill rate, and the BDA rate. One approach would be to assume that kills follow a Poisson process, so the state of a target within a period can be expressed as a continuous-time Markov chain. Unfortunately, any attempt to use this type of model will result in expected numbers of live and dead targets being nonlinear functions of the numbers of targets killed, which are decision variables in the LP. We would welcome research in this area, as it may be possible to build a more rigorous model that can stochastically describe all target states and then develop a linear approximation that will work in an optimization.

In the meantime, the model we have presented does account heuristically for stochastic target regeneration. It allows regeneration within a period, control over the probability of repair, and sortie consumption due to bad BDA, features which are not available in the existing models. It is also a linear function of the optimization's decision variables, keeping the overall model linear.

To describe how TIME STRIKE uses these equations, we will revise the notation somewhat. Let

$\mathbf{n_t}$	number of planning cycles of length **C** in period **t**
$\mathbf{R_t}$	expected number of targets in repair at the end of period **t**
$\mathbf{D_t}$	expected number of dead targets at the end of the period **t**
$\mathbf{K_t}$	total number of (nonrandom) kills allocated by the LP in period **t**

The first step is to determine the repair probabilities. The density function for **X**, the repair time, is $f_X(x) = re^{-rx}$, $x \geq 0$; therefore,

$$P_c = P(0 \leq X \leq C) = \int_0^C re^{-rx} dx$$

$$= 1 - e^{-rC}$$

$$NP_c = 1 - P_c = e^{-rC}$$

We want to know the number of targets dead and in repair at the end of each period to compute progress towards the goals. We can write the expressions for $\mathbf{R}_t$ and $\mathbf{D}_t$ in the following form:

$$R_t = G_t * R_{t-1} - H_{t-1} * K_{t-1} + J_t * K_t$$
$$D_t = D_{t-1} - L_{t-1} * K_{t-1} + M_t * K_t$$

where

$$G_t = NP_c^{\,n_t}$$
$$H_t = \frac{NP_c^{\,n_{t+1}-1} * P_r * NB * (1 + NP_r - P_c)}{n_t}$$
$$J_t = \frac{P_r}{n_t} * \left\{ \frac{[NP_c - NB * (1 + NP_r - P_c)] * (1 - NP_c^{\,n_t - 1})}{P_c} + 1 \right\}$$
$$L_t = \frac{NB * NP_r^2}{n_t}$$
$$M_t = NP_r * \left[1 - NP_r * \left(NB - \frac{NB}{n_t} \right) \right]$$

Expanding the difference equations leads to the following:

$$\begin{aligned} R_t + D_t = {} & J_t * K_t + (G_t * J_{t-1} - H_{t-1}) * K_{t-1} + \\ & \sum_{i=1}^{t-2} \left[\left(\prod_{j=i+2}^{t} G_j \right) * (G_{i+1} * J_i - H_i) * K_i \right] + \\ & M_t * K_t + \sum_{i=1}^{t-1} (M_i - L_i) K_i \end{aligned}$$

In TIME STRIKE, the constants $\mathbf{REPF}_{\mathbf{mkt't}}$ represent the multipliers for targets in repair, and are computed in the GAMS code from constants denoted as $\mathbf{REPG}_{\mathbf{mkt}}$, $\mathbf{REPH}_{\mathbf{mkt}}$, and $\mathbf{REPJ}_{\mathbf{mkt}}$. In the code, $\mathbf{DEADA}_{\mathbf{mkt}}$ and $\mathbf{DEADB}_{\mathbf{mkt}}$ are the multipliers for dead targets:

$$REPF_{mkt't} = \begin{cases} J_{t}, & t' = t \\ G_{t'+1} * J_{t'} - H_{t'}, & t' = t-1 \\ \left(\prod_{j=t'+2}^{t} G_j \right) * \left(G_{t'+1} * J_{t'} - H_{t'} \right), & t' < t-1 \end{cases}$$

$$DEADA_{mkt} = M_t$$

$$DEADB_{mkt} = M_t - L_t$$

Summing these terms gives the total expected proportion of targets struck in time period **t'** that are dead or in repair in time period **t**. While these terms are not combined in the GAMS code, we denote their sum in the model data as $\mathbf{BDAREG_{mkt't}}$ to simply the documentation:

$$BDAREG_{mkt't} = DEADA_{mkt'}\big|_{t'=t} + DEADB_{mkt'}\big|_{t'<t} + REPF_{mkt't}$$

A final caution: TIME STRIKE does not explicitly force restrikes resulting from the last planning cycle of a period to occur in the following period. The reason for this is the model tends to do the restrikes naturally for the first four objectives, and the constraints used with phase-goal objectives require nondecreasing achievement across time and indirectly force restrikes. The only effect of not doing a restrike is that the model can have negative achievement against a target type in a period because it thought some targets were still alive and did not do the restrikes to confirm their status.

Adding explicit constraints would increase the number of rows in the LP by about 30%, and this increase is unwarranted. Failure to do the restrikes is rare in our testing, and the functional outcome in the absence of the constraints is not unreasonable: TIME STRIKE will merely think it achieved a certain level of achievement in a period, and the level will decrease in the next period because the model chose not to schedule the restrikes. As a result, failing to do restrikes makes the model think it has achieved less than it actually has.

APPENDIX B: SOLUTION PROCEDURE FOR PHASE GOALS

The phase-goal objective requires using binary variables to ensure the phases are achieved in hierarchical order. However, we have found it easier to use a simple heuristic to set the values of the binary variables after each continuous LP relaxation than to use branch-and-bound.

The phase goals are designed so that each phase must be achieved to a *threshold*, or proportion of achievement, before the next phase can begin. The continuous relaxation cannot guarantee this behavior, despite all the constraints. As an example, consider the proportions of phase achievement from an initial LP run with m=1, h=2, and t=5, and the threshold set at .8 for all phases. Figure B1 shows the results:

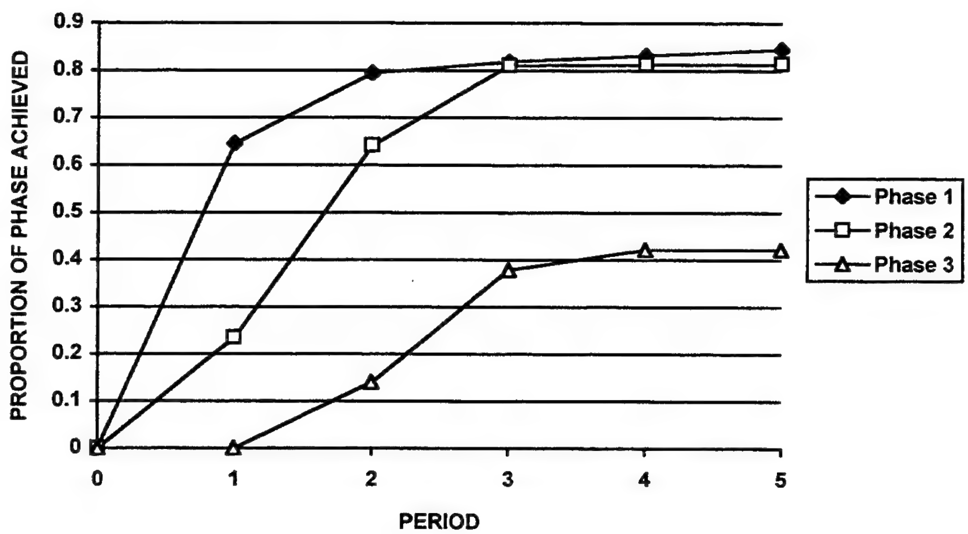

Figure B1. Phase achievement in the initial LP relaxation for the phase-goal objective. This graphic shows what happens in the initial solution. Phase 2 should not start until Phase 1 reaches its threshold achievement of .8, which occurs sometime in period 3. However, the continuous relaxation allows the model to kill targets in Phases 2 and 3 prior to period 3.

The solution of the first continuous LP relaxation with the phase-goal objective shows Phase 2 and Phase 3 targets are being attacked before Phase 1 reaches the threshold. This is not correct; Phase 2 and Phase 3 should not begin before period 3.

The heuristic we use is: first, solve the continuous LP relaxation; second, determine which period the next hierarchical phase achieves its threshold; third, fix the applicable binary variables (the $\mathbf{period}_{mht}$ variables in the formulation) to prevent succeeding phases from starting before that period; fourth, solve the next continuous relaxation for the next phase. The results of this procedure for our example are shown in Figure B2:

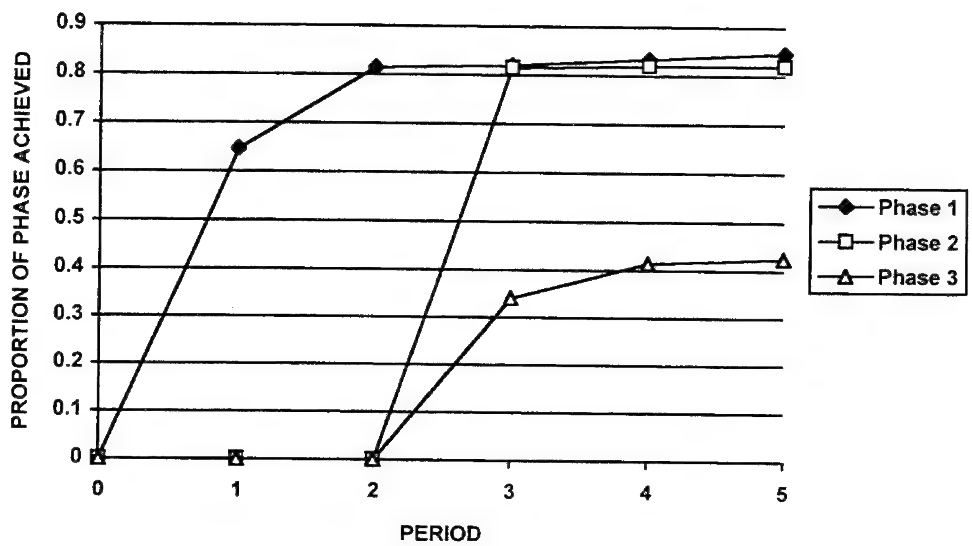

Figure B2. Phase achievement after setting preemptive binary variables. This graphic shows what happens when Phase 2 and Phase 3 kills are prevented before period 3, based on the results of the first LP relaxation. This second, restricted LP reaches a solution that naturally enforces the preemption between Phase 2 and Phase 3, so neither Phase 2 or 3 starts prior to the previous phase reaching the .8 threshold.

This solution also meets the preemption requirements for Phase 3, so no further LP relaxations are required.

In general, TIME STRIKE will require at most **h** LP relaxations for **h** phases. This procedure is completely automated in the GAMS code, and requires no manual intervention by the user. Since GAMS passes the basis from the previous solution to the next, restricted solution, the solution times tend to be very quick. A large 2-MRC problem that required 18 minutes for an initial solution will solve the succeeding phases in less than 1 minute each. Of course, we would expect this, because only one variable is being fixed at a bound after each relaxation.

These diagrams also illustrate the importance of setting a threshold. In this instance of TIME STRIKE, a threshold of 1.0 for Phase 1 (pure preemption) would mean *nothing* in Phases 2 or 3 could ever be attacked, because the model never completely achieves Phase 1. Our testing has shown that this is the most common case; due to restrikes and target regeneration, the model can rarely achieve more than 97-99% of a goal. Nonetheless, we argue that this is realistic, and campaign phases tend to overlap in modern air warfare. It's highly unlikely a commander would stop the campaign because of the inability to kill a handful of targets.

However, testing has shown setting high thresholds is a useful analysis technique. If a user is trying to determine which target types and phase goals are influencing the overall solution, setting high thresholds will uncover them quickly.

We have experimented with solving the phase-goal formulation as a pure mixed-integer program (MIP) using branch-and-bound techniques. Since different solvers offer different branching strategies, getting good integer solutions quickly requires experimentation with the solver. Since the first LP relaxation provides a lower bound on the overall solution, a good indication of whether branch-and-bound is justified is whether the objective from the first relaxation is substantially different from the objective after the final pass in the heuristic. In our testing, the differences have been small, with an integrality gaps of less than 10%. However, branch-and-bound has performed better than the heuristic for some test problems. As a result, TIME STRIKE contains an option to solve the problem as a MIP, but users should note that they should be prepared to spend some time "tuning" their solver's branch-and-bound strategies to get solutions in a reasonable amount of time.

APPENDIX C: DERIVATION OF TSORT

TIME STRIKE must account for aircraft attrition within a period. Unfortunately, attrition is a function of the sortie assignments, so the number of available sorties is a function of the sortie rate, the attrition rate, and the length of the time period (denoted by **s**, **a**, and **t** respectively). In the formulation, $\mathbf{TSORT_{mijkpt}}$ is the upper bound on the number of possible sorties a single aircraft could generate for the particular combination; the following is a derivation of this bound from a stochastic point of view.

Let **TS** be the random variable representing the number of sorties flown in a period. We want the expected value **E(TS)**, which we will find by conditioning on the number of sorties flown.

Suppose the maximum number of sorties than can be flown in a given interval is denoted by the random variable **Y**. Then, the conditional probability distribution of **TS** given **Y** is a truncated geometric distribution:

$$P(TS = x|Y = y) = \begin{cases} a(1-a)^{x-1}, & 1 \le x < y \\ (1-a)^{y-1}, & x = y \end{cases}$$

The expected value **E(TS|Y=y)** is as follows. For convenience, let **q=1-a**:

$$\begin{aligned} E(TS|Y = y) &= \sum_{x=1}^{y-1} xaq^{x-1} + yq^{y-1} \\ &= a\sum_{x=1}^{y-1} \frac{d}{dq}q^{x} + yq^{y-1} \\ &= a\frac{d}{dq}\left(\sum_{x=1}^{y-1} q^{x}\right) + yq^{y-1} \end{aligned}$$

Using an identity for the partial sum of a geometric series, we can find a closed-form expression for the sum:

$$\sum_{x=1}^{y-1} q^{x-1} = \frac{1-q^{y-1}}{1-q}, \text{ so}$$

$$\sum_{x=1}^{y-1} q^{x} = \frac{1-q^{y-1}}{1-q} - q^0 + q^{y-1}$$

After taking the derivative and reducing, we have

$$\begin{aligned}
E(TS|Y=y) &= \frac{1-q^{y-1}+aq^{y-1}}{a} \\
&= \frac{1-(1-a)q^{y-1}}{a} \\
&= \frac{1-q^{y}}{a} \\
&= \frac{1-(1-a)^{y}}{a}
\end{aligned}$$

This is the formula used in both TAM and MIXMASTER, with **st** used in place of y. However, if **y** is not an integer, this formula does not hold. To get to the result used in TIME STRIKE, instead assume that sorties are flown according to a Poisson process with rate s. Then, the distribution of **Y** is Poisson with mean **st**. We use this to uncondition **E(TS|Y)**:

$$\begin{aligned}
E(TS) &= E\left[E(TS|Y)\right] \\
&= \sum_{y=0}^{\infty} \frac{1-(1-a)^{y}}{a} * P(Y=y) \\
&= \sum_{y=0}^{\infty} \frac{1-(1-a)^{y}}{a} * \frac{e^{-st}(st)^{y}}{y!} \\
&= \frac{1}{a}\left[\sum_{y=0}^{\infty} \frac{e^{-st}(st)^{y}}{y!} - \sum_{y=0}^{\infty}(1-a)^{y} * \frac{e^{-st}(st)^{y}}{y!}\right] \\
&= \frac{1}{a}\left[\sum_{y=0}^{\infty} \frac{e^{-st}(st)^{y}}{y!} - \frac{e^{-st}}{e^{-st+ast}} \sum_{y=0}^{\infty} \frac{e^{-st+ast}(st-ast)^{y}}{y!}\right]
\end{aligned}$$

Both sums are Poisson density functions, so both sum to 1. This leaves the final result:

$$E(TS) = \frac{1}{a}\left[1 - \frac{e^{-st}}{e^{-st+ast}}\right]$$
$$= \frac{1 - e^{-ast}}{a}$$

Converting to the data definitions used in TIME STRIKE, we have

$$TSORT_{mijkpt} = \frac{1 - e^{-ATTR_{mijkpt} * SR_{mit} * NDAYS_t}}{ATTR_{mijkpt}}$$

Note that we can also get this result by using differential equations. Let $\mathbf{A_t}$ be the fraction of a single aircraft left at time **t**. Then,

$$\frac{dA_t}{dt} = -A_t as$$

This assumes the aircraft decays as a function of the attrition rate, the sortie rate, and the proportion of the aircraft remaining. This can be solved using separation of variables:

$$\frac{dA_t}{A_t} = -asdt$$
$$\int \frac{dA_t}{A_t} = \int -asdt$$
$$\ln|A_t| = -ast + c$$
$$A_t = e^{-ast+c} = ke^{-ast}$$

At **t** = 0 we have one airplane, so **k** = 1. Integrating using **s** to determine the total number of sorties leads to the result:

$$\int_o^t se^{-asx}dx = s\left[-\frac{1}{as}e^{-asx}\right]_0^t = \frac{1-e^{-ast}}{a}$$

LIST OF REFERENCES

Boger, Dan C., and Alan R. Washburn, *Notes from the Stockpile Seminar*, Naval Postgraduate School, Monterey, CA, August 1985.

Brooke, Anthony, David Kendrick, and Alexander Meeraus, *GAMS: A User's Guide*, Danvers, MA, Boyd and Fraser, 1992.

Brown, Gerald G., Alan R. Washburn, and Dennis M. Coulter, "Sortie Optimization and Munitions Planning," *Military Operations Research*, Vol. 1, No. 1, Summer 1994, pp. 13-18.

Clausen, R. J., G. W. Graves, and J. Y. Lu, *Sortie Allocation by a Nonlinear Programming Model for Determining a Munitions Mix*, The RAND Corporation, R-1411-DDPAE, March 1974.

DeYonke, John R., *A Note on the Formulation of the QUICK STRIKE Model*, Working Paper, Office of Aerospace Studies, Kirtland AFB, NM, September 1995.

Jackson, Jack A. Jr., *A Taxonomy of Advanced Linear Programming Techniques and the Theater Attack Model*, thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, March 1989.

Lord, Paul H., *An Examination of the United States Air Force Optimal Nonnuclear Munitions Procurement Model*, thesis, Naval Postgraduate School, Monterey, CA, October 1982.

Langbehn, Skip, and K. Lindsey, *Munitions Roadmap Study*, briefing, HQ ACC/XP-JSG, Langley AFB, VA, 25 January 1991.

Might, Robert J., "Decision Support for Aircraft and Munitions Procurement," *Interfaces*, Vol. 17, No. 5 (September-October 1987), pp. 55-63.

Ross, Sheldon R., *Introduction to Probability Models*, Academic Press, Inc., 1993.

Yost, Kirk A., *Survey and Description of USAF Conventional Munitions Allocation Models*, OAS TR 95-1, Office of Aerospace Studies, Kirtland AFB, NM, 10 February 1995.

INITIAL DISTRIBUTION LIST

No. of copies	
1	Research Office (Code 08) Naval Postgraduate School Monterey, CA 93943-5002
2	Dudley Knox Library (Code 52) Naval Postgraduate School Monterey, CA 93943-5002
2	Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218
50	Department of Operations Research Naval Postgraduate School Monterey, CA 93943-5002
1	Professor. Alan Washburn Code OR/WS Naval Postgraduate School Monterey, CA 93493-5000
1	Professor. Jerry Brown Code OR/BW Naval Postgraduate School Monterey, CA 93493-5000
1	Professor Robert Dell Code OR/DE Naval Postgraduate School Monterey, CA 93943-5000
1	AEM Services, Inc. Attn: Mr. Bill Cotsworth 8480 E. Orchard Road Greenwood Village, CO 80111
2	AFIT/ENS Attn: Dr. James Chrissis/ LtCol Jack Jackson Wright-Patterson AFB, OH 45433-6583
4	AFSAA/SAG/SAS/SAZ/SAMI 1570 Air Force Pentagon Washington, DC 20330-1570

No. of copies	
1	ASC/XREC Attn: Mr. Frank Campanile 1970 3rd Street, Suite 2, Bldg 11A Wright-Patterson AFB, OH 45433- 7209
1	ASC/XREW 101 W. Eglin Blvd, Suite 388 Eglin AFB, FL 32542-5499
1	ASI Systems International Eglin Group 838 N. Eglin Parkway, Suite 421 Fort Walton Beach, FL 32547-2592
1	ASI Systems International 5203 Leesburg Pike, Suite 1602 Falls Church, VA 22041
1	CADRE/EDW 401 Chennault Circle Maxwell AFB, AL 36112-6428
1	Cambridge Research Associates Attn: Dr. Greg Burgess 1430 Spring Hill Rd McLean, VA 22102
1	Center For Naval Analyses Attn: Dr. Ron Nickel 4401 Ford Ave. Alexandria, VA 22302-0268
1	COMOPTEVFOR Attn: Dr. Jim Duff 7970 Diven Street Norfolk, VA 23505-1498
1	Concepts Analysis Agency Attn: Mr. E. B. Vandiveer 8120 Woodmont Avenue Bethesda, MD 20814-2797

1 Field Command, Defense Nuclear Agency
Attn: Capt Skip Langbehn
1680 Texas Street SE
Kirtland AFB, NM 87117-5669

1 George Mason University
Department of Systems Engineering
Attn: Dr. Bob Might
Fairfax, VA 22030-4444

2 HQ ACC/XP-SAS
Attn: Mr. Rich Freet/Maj Glen Ferrar
204 Dodd Blvd, Suite 202
Langley AFB, VA 23665-2778

1 HQ USA/DCS-OPS
Attn: Mr. John Riente
400 Army
Pentagon
Washington, DC 20310-0400

1 HQ USAF/XOFW
1480 Air Force Pentagon, Room 4A1084
Washington, DC 20330-1480

2 HQ USAF/XOMT/XOME
1480 Air Force Pentagon
Washington, DC 20330-1480

2 HQ USAF/XOOC
Attn: Maj Jim Hutto
1520 Air Force, Pentagon, Rm B922B
Washington, DC 20330-1520

2 Institute for Defense Analyses
Attn: Dr. Jim Bexfield
1801 N. Beauregard Street
Alexandria, VA 22311

1 JAST/PIA
Attn: Lt Col Kenneth C. Konwin
Cystal Square 4, Suite 307
1745 Jefferson-Davis Highway
Arlington, VA 22202

2 JCS J-8
CFAD
Attn: Maj Bob O'Neill
Pentagon, Room 1D963
Washington, DC 20318-8000

1 Logistics Management Institute
Attn: Mr. Sal Culosi
2000 Corporate Ridge
McLean, VA 22102-7805

1 McDonnell Douglas Aerospace
Attn: Mr. Richard Morris
1735 Jefferson Davis Highway, 11th Floor
Arlington, VA 22202

1 Naval Air Systems Command, AIR 4.10
Attn: Mr. Ralph Mosios
1421 Jefferson Davis Highway
Arlington, VA 22243-5260

2 OASD (PA&E)
Attn: Dr. Jackie Henningsen/
Mr. Roy Hempley
Pentagon
Washington, DC 20301-1800

22 OAS/DR
3550 Aberdeen Ave
Kirtland AFB, NM 87117-5776

1 OPNAV/N8
N184J (NNOR)
Attn: Mr. Paul Hubai
Pentagon, Room 4A478
Washington, DC 20330

2 PL/HO
Kirtland AFB, NM 87117-5000

1 Quantics, Inc.
Attn: Mr. Bruce Backup
801 Springdale Drive
Exton, PA 19341

1 RAND
Attn: Dr. John Friel
1700 Main Street
PO Box 2138
Santa Monica, CA 90407-2138

1 S3I
Attn: Mr. Damon Lum
1800 Diagonal Road
Suite 140
Alexandria, VA 22314

1 Science Applications International
Attn: Mr. James Sebolka
8201 Greensboro Drive, Suite 470
McLean, VA 22102

1 Sverdrup Technologies
Attn: Mr. Darryl Thornton
PO Box 1935
Eglin AFB, Fl 32542

1 USAFA/DFMS
Attn: Director of Research
USAF Academy, CO 80840-6252

1 VEDA, Inc.
Attn: Dr. Kirk Sturtz
5200 Springfield Street, Suite 200
Dayton, OH 45431

1 Dr. Roy Rice
Teledyne-Brown Engineering
P.O. Box 070007
Huntsville, AL 35807-7007

2 Dr. Neal D. Glassman
AFOSR/NM
110 Duncan Avenue, Suite 100
Bolling AFB, Washington, D.C.
20332-001

2 Dr. Donald K. Wagner
Office of Naval Research
Div. of Math Sciences, Code 311
800 N. Quincy Avenue
Arlington, VA 22217-5660